ONVIF™

Profile G Client Test Specification

Version 16.07

July 2016

REVISION HISTORY

Vers.	Date	Description
16.07	Apr 20, 2016	• Test cases about specific event were removed: RECORDINGCONTROL-7, RECORDINGCONTROL-8, RECORDINGCONFIGURATION-5, RECORDINGCONFIGURATION-6, TRACKCONFIGURATION-3, RECEIVER-8, RECEIVER-9.
16.07	Apr 18, 2016	Step description in Test Procedure was updated for the test cases: REPLAYCONTROL-2, REPLAYCONTROL-3, REPLAYCONTROL-4. Old description: Device response has code RTSP 200 OK if it is detected New description: If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK
16.07	Mar 18, 2016	Checking of TEARDOWN response was changed in Test Procedure and PASS criteria for the test cases and annexes: REPLAYCONTROL-2, REPLAYCONTROL-3, REPLAYCONTROL-4 Old description of checking of TEARDOWN response in Test Procedure: Device responds with code RTSP 200 OK. New description of checking of TEARDOWN response in Test Procedure: Device response has code RTSP 200 OK if it is detected. Old description of checking of TEARDOWN response in PASS criteria: [S32] Device response contains "RTSP/* 200 OK" New description of checking of TEARDOWN response in PASS criteria: [S32] Device response contains "RTSP/* 200 OK" if it is detected
16.07	Mar 16, 2016	Docbook stylesheets were updated.
16.07	Mar 14, 2016	www.onvif.org was removed from Copyright section.
16.07	Mar 09, 2016	REPLAYCONTROL-5 test case: Profile G Requirement was changed from Conditional to Optional.
16.07	Jan 28, 2016	RFC 2326 was added to normative reference. The description about structure and hierarchy was replaced for the test cases: MEDIASEARCH-1, MEDIASEARCH-2, MEDIASEARCH-3, MEDIASEARCH-4, MEDIASEARCH-5, MEDIASEARCH-6, REPLAYCONTROL-1, REPLAYCONTROL-2, REPLAYCONTROL-3, REPLAYCONTROL-4, REPLAYCONTROL-6 Old description:

Client %COMMAND NAME% request message is a well-formed SOAP request (refer to onvif.xsd) AND

Client %COMMAND NAME% request message has a proper hierarchy (refer to %SERVICE%.wsdl) AND

New description:

Client %COMMAND NAME% request messages are valid according to XML Schemas listed in Namespaces AND

Client %COMMAND NAME% request in Test Procedure fulfills the following requirements:

The following steps was removed because the requirements are fullfield by XML Schemas validation:

- MEDIASEARCH-1:

  [S2] "<FindRecordings>" includes tag: "<Scope>" AND

  [S3] "<FindRecordings>" includes tag: "<KeepAliveTime>" AND

  [S7] "<GetRecordingSearchResults>" includes tag: "<SearchToken>" AND
- MEDIASEARCH-2:

  [S2] "<FindEvents>" includes tag: "<StartPoint>" AND

  [S3] "<FindEvents>" includes tag: "<Scope>" AND

  [S4] "<FindEvents>" includes tag: "<SearchFilter>" AND

  [S5] "<FindEvents>" includes tag: "<IncludeStartState>" AND

  [S6] "<FindEvents>" includes tag: "<KeepAliveTime>" AND

  [S10] "<GetEventSearchResults>" includes tag: "<SearchToken>" AND
- MEDIASEARCH-5:

  [S2] "<GetMediaAttributes>" includes tag: "<Time>" AND
- MEDIASEARCH-6:

  [S2] "<FindEvents>" includes tag: "<StartPoint>" AND

  [S3] "<FindEvents>" includes tag: "<Scope>" AND

  [S6] "<FindEvents>" includes tag: "<IncludeStartState>" AND

  [S7] "<FindEvents>" includes tag: "<KeepAliveTime>" AND

  [S4] "<FindEvents>" includes tag: "<SearchFilter>" AND
- REPLAYCONTROL-1:

  [S3] "<StreamSetup>" includes tag: "<Stream>" with ("RTP-unicast" OR "RTP-multicast") value AND

  [S2] "<GetReplayUri>" includes tag: "<StreamSetup>" AND

  [S4] "<StreamSetup>" includes tag: "<Transport>" AND
- REPLAYCONTROL-2:

  [S2] "<GetReplayUri>" includes tag: "<StreamSetup>" AND

		• REPLAYCONTROL-3: [S2] "<GetReplayUri>" includes tag: "<StreamSetup>" AND • REPLAYCONTROL-4: [S2] "<GetReplayUri>" includes tag: "<StreamSetup>" AND • REPLAYCONTROL-6: [S2] "<SetReplayConfiguration>" includes tag: "<Configuration>" AND [S3] "<Configuration>" includes tag: "<SessionTimeout>" with non-empty value of duration AND
16.01	Nov 25, 2015	General item (Test Overview) was added Minor updates in formatting, typos and terms Updates according review results: • Dynamic Recording Configurations test cases • Dynamic Tracks Configurations test cases • Recording Control test cases • Recording Configuration test cases • Track Configuration test cases • Recording Control – Using a Receiver as Source test cases
16.01	Sep 23, 2015	Dynamic Recording Configurations test cases added Dynamic Tracks Configurations test cases added Recording Control test cases added Namespaces section added Recording Configuration test cases added Track Configuration test cases added Recording Control – Using a Receiver as Source test cases added
15.06	Jun 10, 2015	No major changes were made, just minor formatting fixes.
15.05	May 20, 2015	No major changes were made, just minor grammatical corrections.
15.02	Feb 19, 2015	Pass criteria in MEDIASEARCH-6 test case has been updated (removed unnecessary checks of <SearchFilter> tag value).
14.12	Dec 19, 2014	Added Note to Test Result section of MEDIASEARCH-6 Test Case.
14.12	Dec 11, 2014	Fixed typos and inconsistencies.
1.0	Oct 17, 2014	Initial version

Table of Contents

1 Introduction

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

This particular document defines test cases required for testing Profile G features of a Client application e.g. Recording Search - Media Search. 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 G Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile G 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 G features according to ONVIF Profile Specifications.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Profile G 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 Recording Search - Media Search

Recording Search - Media Search section specifies Client ability to perform operations for finding data of interest within a set of recordings or events on Device.

1.3 Replay Control

Replay Control section specifies Client ability to control replay of stored video, audio and metadata on Device. This section also specifies Client ability to configure RTSP session timeout.

1.4 Recording Control – Dynamic Recording Configurations

Recording Control – Dynamic Recording Configurations section specifies Client ability to create and delete recordings on Device.

1.5 Recording Control – Dynamic Tracks Configurations

Recording Control – Dynamic Tracks Configurations section specifies Client ability to create and delete tracks on Device.

1.6 Recording Control

Recording Control section specifies Client ability listing of recordings, managing recording jobs, and managing the state of a recording job on Device. This section also specifies Client ability to retrieve notifications of the change in a recording job's state.

1.7 Recording Configuration

Recording Configuration section specifies Client ability get recording configuration, change recording configuration, get recording job configuration, change recording job configuration for Device. This section also specifies Client ability to retrieve notifications of recording configuration change and recording job's configuration change events.

1.8 Track Configuration

Track Configuration section specifies Client ability get track configuration, change track configuration for Device. This section also specifies Client ability to retrieve notifications of track configuration change events.

1.9 Recording Control – Using a Receiver as Source

Recording Control – Using a Receiver as Source section specifies Client ability listing of receivers, managing receivers, and managing the state and mode of a receivers on Device. This section also specifies Client ability to retrieve notifications of the change in a receivers state and connection failed.

2 Normative references

- ONVIF Conformance Process Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Profile Policy:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Core Specifications:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Core Client Test Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Profile G Specification version 1.0, Jun 2014:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Recording Search Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Recording Control Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Replay Control Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Receiver Specification:

  http://www.onvif.org/Documents/Specifications.aspx
- ONVIF Streaming Specification:

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

- W3C Web Services Addressing 1.0 – Core:

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

- ONVIF Network Interface Specification Set:

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

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

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

3 Terms and Definitions

3.1 Conventions

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

3.2 Definitions

This section describes terms and definitions used in this document.

Profile	See ONVIF Profile Policy.
Profile G	The Profile G 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.
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).
Recording	A container for a set of audio, video and metadata tracks. A recording can hold one or more tracks. A track is viewed as an infinite timeline that holds data at certain times.
Track	An individual data channel consisting of video, audio, or metadata. This definition is consistent with the definition of track in [RFC 2326].
Video Analytics	Algorithms or programs used to analyze video data and to generate data describing object location and behaviour.
Valid Device Response	Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.

3.3 Abbreviations

This section describes abbreviations used in this document.

HTTP Hyper Text Transport Protocol.

HTTPS Hyper Text Transport Protocol over Secure Socket Layer.

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]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML schema instance namespace
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
tds	http://www.onvif.org/ver10/device/wsdl	The namespace for the WSDL device service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
trv	http://www.onvif.org/ver10/receiver/wsdl	The namespace for the WSDL receiver service
trc	http://www.onvif.org/ver10/recording/wsdl	The namespace for the WSDL recording service
tse	http://www.onvif.org/ver10/search/wsdl	The namespace for the WSDL search service
trp	http://www.onvif.org/ver10/replay/wsdl	The namespace for the WSDL replay service
ter	http://www.onvif.org/ver10/error	The namespace for ONVIF defined faults

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 the Profile G is an ONVIF Client that can configure, request, and control recording of video data over an IP network from an ONVIF Device compliant to the Profile G. The Profile G also includes support for receiving audio and metadata stream if the Client supports those features.

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

4.1 General

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

- Feature Level Requirement
- Expected Scenarios Under Test
- List of Test Cases

4.1.1 Feature Level Requirement

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

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

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

4.1.2 Expected Scenarios Under Test

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

4.1.3 Test Cases

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

Each Test Case contains the following parts:

- Test Label - Unique label for each test
- Test Case ID - Unique ID for each test
- Profile Normative References - Requirement level for the feature under test is defined in Profile Specification. This reference is informative and will not be used in conformance procedure.
- Feature Under Test - Feature which is under current test. Typically a particular command or an event.
- Test Purpose - The purpose of current test case.
- Pre-Requisite - The pre-requisite defines when the test should be performed. In case if pre-reqiusite does not match, the test result will be NOT DETECTED.
- Test Procedure - scenario expected to be reflected in network trace file.
- Test Result - Passed and failed criteria of the test case. Depending on these criteria test result will be defined as PASSED or FAILED.
- 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 G, 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 Recording Search - Media Search Test Cases

5.1 Expected Scenarios Under Test:

1. Client connects to Device to start a search session.
2. Client is considered as supporting Recording Search - Media Search if the following conditions are met:
   - Client is able to perform Recording Search using FindRecordings and GetRecordingSearchResults operations AND
   - Client is able to perform Event Search using FindEvents and GetEventSearchResults operations.
3. Client is considered as NOT supporting Recording Search - Media Search if ANY of the following is TRUE:
   - No valid responses for FindRecordings OR
   - No valid responses for GetRecordingSearchResults OR
   - No valid responses for FindEvents OR
   - No valid responses for GetEventSearchResults
4. If applicable for Client then any of the following conditions shall be met:
   - Client is able to retrieve Recording Summary using GetRecordingSummary operation OR
   - Client is able to retrieve Recording Information using GetRecordingInformation operation OR
   - Client is able to retrieve Media Attributes using GetMediaAttributes operation OR
   - Client is able to set a search filters using XPath dialect expressions (e.g. for FindEvents operation).

5.2 RECORDING SEARCH

Test Label: Media Search - Search for Recordings on Device

Test Case ID: MEDIASEARCH-1

Profile G Normative Reference: Mandatory

Feature Under Test: Media Search

Test Purpose: To verify that the Client is able to perform recordings search session using FindRecordings and GetRecordingSearchResults operations.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with FindRecordings and GetRecordingSearchResults operations present.

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

1. Client invokes FindRecordings request message to starts a search session, looking for recordings that matches the scope.
2. Device responds with code HTTP 200 OK and FindRecordingsResponse message.
3. Client invokes GetRecordingSearchResults request message to receive the results from a recording search session previously initiated by a FindRecordings operation.
4. Device responds with code HTTP 200 OK and GetRecordingSearchResultsResponse message.

Test Result:

PASS -

- Client **FindRecordings** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **FindRecordings** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<FindRecordings>" tag after the "<Body>" tag AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<FindRecordingsResponse>" tag AND
- Client **GetRecordingSearchResults** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetRecordingSearchResults** request in Test Procedure fulfills the following requirements:
  - [S6] Client request contains "<GetRecordingSearchResults>" tag after the "<Body>" tag AND
  - [S8] Device response contains "HTTP/* 200 OK" AND

- [S9] Device response contains "<GetRecordingSearchResultsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_RecordingSearch

5.3 EVENT SEARCH

Test Label: Media Search - Search for Events on Device

Test Case ID: MEDIASEARCH-2

Profile G Normative Reference: Mandatory

Feature Under Test: Media Search

Test Purpose: To verify that the Client is able to perform events search session using FindEvents and GetEventSearchResults operations.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with FindEvents and GetEventSearchResults operations present.

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

1. Client invokes FindEvents request message to starts a search session, looking for events that matches the search scope.
2. Device responds with code HTTP 200 OK and FindEventsResponse message.
3. Client invokes GetEventSearchResults request message to receive the results from a recording search session previously initiated by a FindEvents operation.
4. Device responds with code HTTP 200 OK and GetEventSearchResultsResponse message.

Test Result:

PASS -

- Client **FindEvents** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **FindEvents** request in Test Procedure fulfills the following requirements:

- [S1] Client request contains "<FindEvents>" tag after the "<Body>" tag AND
- [S7] Device response contains "HTTP/* 200 OK" AND
- [S8] Device response contains "<FindEventsResponse>" tag AND

- Client **GetEventSearchResults** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetEventSearchResults** request in Test Procedure fulfills the following requirements:
  - [S9] Client request contains "<GetEventSearchResults>" tag after the "<Body>" tag AND
  - [S11] Device response contains "HTTP/* 200 OK" AND
  - [S12] Device response contains "<GetEventSearchResultsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_EventSearch

5.4 GET RECORDING SUMMARY

Test Label: Media Search - Get Recording Summary

Test Case ID: MEDIASEARCH-3

Profile G Normative Reference: Conditional

Feature Under Test: Media Search

Test Purpose: To verify that Client is able to retrieve Recording Summary using GetRecordingSummary operation.

Pre-Requisite:

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

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

1. Client invokes GetRecordingSummary request message to get a summary description of all recorded data.
2. Device responds with code HTTP 200 OK and GetRecordingSummaryResponse message.

Test Result:

PASS -

- Client **GetRecordingSummary** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetRecordingSummary** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetRecordingSummary>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetRecordingSummaryResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_RecordingSummary

5.5 GET RECORDING INFORMATION

Test Label: Media Search - Get Recording Information

Test Case ID: MEDIASEARCH-4

Profile G Normative Reference: Conditional

Feature Under Test: Media Search

Test Purpose: To verify that Client is able to retrieve Recording Information using GetRecordingInformation operation.

Pre-Requisite:

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

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

1. Client invokes GetRecordingInformation request message to retrieve information about a single Recording specified by a RecordingToken.
2. Device responds with code HTTP 200 OK and GetRecordingInformationResponse message.

Test Result:

PASS -

- Client **GetRecordingInformation** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetRecordingInformation** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetRecordingInformation>" tag after the "<Body>" tag AND
  - [S2] "<GetRecordingInformation>" includes tag: "<RecordingToken>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetRecordingInformationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_RecordingInformation

5.6 GET MEDIA ATTRIBUTES

Test Label: Media Search - Get Media Attributes

Test Case ID: MEDIASEARCH-5

Profile G Normative Reference: Conditional

Feature Under Test: Media Search

Test Purpose: To verify that Client is able to retrieve a set of media attributes using GetMediaAttributes operation.

Pre-Requisite:

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

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

1. Client invokes GetMediaAttributes request message to retrieve a set of media attributes for all tracks of the specified recordings at a specified point in time.
2. Device responds with code HTTP 200 OK and GetMediaAttributesResponse message.

Test Result:

PASS -

- Client **GetMediaAttributes** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMediaAttributes** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetMediaAttributes>" tag after the "<Body>" tag AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetMediaAttributesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_MediaAttributes

5.7 FIND EVENTS WITH SEARCH FILTERS

Test Label: Media Search - SearchFilter specified in FindEvents

Test Case ID: MEDIASEARCH-6

Profile G Normative Reference: Conditional

Feature Under Test: Media Search

Test Purpose: To verify that the Client is able to set a search filters using XPath dialect expressions for FindEvents operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with specified SearchFilter element inside FindEvents request message.

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

1. Client invokes FindEvents request message to start a search session with specified SearchFilter element that contains the topic and message filter needed to define what events to search for.
2. Device responds with code HTTP 200 OK and FindEventsResponse message.

Test Result:

NOTE: If Client FindEvents request message does not contain any value in "<SearchFilter>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **FindEvents** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **FindEvents** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<FindEvents>" tag after the "<Body>" tag AND
  - [S5] "<SearchFilter>" contains any XPath expression AND
  - [S8] Device response contains "HTTP/* 200 OK" AND
  - [S9] Device response contains "<FindEventsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: MediaSearch_EventSearchFilter

6 Replay Control Test Cases

6.1 Expected Scenarios Under Test:

1. Client connects to Device to control replay of stored video, audio and metadata.
2. Client is considered as supporting Replay Control if the following conditions are met:
   - Device returns a valid response to GetReplayUri request AND
   - Client is able to initiate playback of a recorded stream from Device with either of the following encoding types:
     - MJPEG OR
     - MPEG4 OR
     - H264
   - When Device and Client support reverse playback capability for media streaming:
     - Client is able to initiate playback using the negative value of Scale header field in RTSP PLAY command.
   - When Device and Client support configurable RTSP session timeout value of the replay service:
     - Client is able to change the value using SetReplayConfiguration operation AND
     - Client is able to retrieve current value using GetReplayConfiguration operation.
3. Client is considered as NOT supporting Replay Control if ANY of the following is TRUE:
   - No Valid Device Response to GetReplayUri request OR
   - Client is unable to initiate playback of a recorded stream from Device
   - When Device and Client support reverse playback capability for media streaming:
     - Client is unable to get valid Device response to RTSP PLAY command with negative value of Scale header field.
   - When Device and Client support configurable RTSP session timeout value of the replay service:
     - Client is unable to get valid Device response to SetReplayConfiguration operation OR

- Client is unable to get valid Device response to GetReplayConfiguration operation.

6.2 GET REPLAY URI

Test Label: Replay Control - Get Replay Uri

Test Case ID: REPLAYCONTROL-1

Profile G Normative Reference: Mandatory

Feature Under Test: Replay Control

Test Purpose: To verify that recorded stream URI from Device is received by Client using the GetReplayUri operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetReplayUri operation present.

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

1. Client invokes GetReplayUri request message with the following Stream Setup: Stream type element with "RTP-unicast" OR "RTP-multicast" value and Transport Protocol element with "UDP" OR "HTTP" OR "RTSP" value and Recording Token element (indicates the media record selected for replay).
2. Device responds with code HTTP 200 OK and GetReplayUriResponse message.

Test Result:

PASS -

- Client **GetReplayUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetReplayUri** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetReplayUri>" tag after the "<Body>" tag AND
  - [S5] "<Transport>" includes tag: "<Protocol>" with ("UDP" OR "HTTP" OR "RTSP") value AND
  - [S6] "<GetReplayUri>" includes tag: "<RecordingToken>" with non-empty string value of specified record token AND
  - [S7] Device response contains "HTTP/* 200 OK" AND

- [S8] Device response contains "<GetReplayUriResponse>" tag

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_GetReplayUri

6.3 MJPEG REPLAY RECORDING

Test Label: Replay Control - MJPEG Replay Recording

Test Case ID: REPLAYCONTROL-2

Profile G Normative Reference: Mandatory

Feature Under Test: Replay Control

Test Purpose: To verify that Client is able to replay stored recording from Device by using GetReplayUri operation and RTSP commands to establish and then stop media stream with MJPEG encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetReplayUri operation and RTSP DESCRIBE, RTSP SETUP, RTSP PLAY and RTSP TEARDOWN commands present.

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

1. Client invokes GetReplayUri request message to retrieve URI of stored recording.
2. Device responds with code HTTP 200 OK and GetReplayUriResponse 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 MIME Type: JPEG.
5. Client invokes RTSP SETUP request with transport parameter element to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes RTSP PLAY request 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.

Test Result:

NOTE: If Device RTSP DESCRIBE response message does not contain Media Type: "video" OR MIME Type: "JPEG" inside SDP information then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **GetReplayUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetReplayUri** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetReplayUri>" tag after the "<Body>" tag AND
  - [S3] "<GetReplayUri>" includes tag: "<RecordingToken>" with non-empty string value of specified record token AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<GetReplayUriResponse>" tag AND
  - [S6] "<GetReplayUriResponse>" includes tag: "<Uri>" with valid URI address AND
  - [S7] Client request introduces RTSP DESCRIBE command AND
- Client RTSP DESCRIBE request command has a proper hierarchy (see [RFC 2326]) AND
  - [S8] RTSP DESCRIBE includes: URI address obtained from GetReplayUriResponse AND
  - [S9] RTSP DESCRIBE includes: "RTSP/*" version identifier AND
  - [S10] RTSP DESCRIBE includes: "CSeq" identifier AND
  - [S11] Device response contains "RTSP/* 200 OK"
  - [S12] Device response SDP information contains Media Type: "video" and MIME Type: "JPEG" AND
  - [S13] Client request introduces RTSP SETUP command AND
- Client RTSP SETUP request command has a proper hierarchy (see [RFC 2326]) AND
  - [S14] RTSP SETUP includes: URI address AND
  - [S15] RTSP SETUP includes: "RTSP/*" version identifier AND

- [S16] RTSP SETUP includes: "CSeq" identifier AND
- [S17] RTSP SETUP includes: "Transport" parameter AND
- [S18] RTSP SETUP includes: "Require" parameter with "onvif-replay" value AND
- [S19] Device response contains "RTSP/* 200 OK" AND
- [S20] Client request introduces RTSP PLAY command AND

- Client RTSP PLAY request command has a proper hierarchy (see [RFC 2326]) AND
  - [S21] RTSP PLAY includes: URI address AND
  - [S22] RTSP PLAY includes: "RTSP/*" version identifier AND
  - [S23] RTSP PLAY includes: "CSeq" identifier AND
  - [S24] RTSP PLAY includes: "Session" parameter AND
  - [S25] RTSP PLAY includes: "Require" parameter with "onvif-replay" value AND
  - [S26] Device response contains "RTSP/* 200 OK" AND
  - [S27] Client request introduces RTSP TEARDOWN command AND
- Client RTSP TEARDOWN request command has a proper hierarchy (see [RFC 2326]) AND
  - [S28] RTSP TEARDOWN includes: URI address AND
  - [S29] RTSP TEARDOWN includes: "RTSP/*" version identifier AND
  - [S30] RTSP TEARDOWN includes: "CSeq" identifier AND
  - [S31] RTSP TEARDOWN includes: "Session" parameter AND
  - [S32] Device response contains "RTSP/* 200 OK" if it is detected.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_MJPEGReplayRecording

6.4 MPEG4 REPLAY RECORDING

Test Label: Replay Control - MPEG4 Replay Recording

Test Case ID: REPLAYCONTROL-3

Profile G Normative Reference: Mandatory

Feature Under Test: Replay Control

Test Purpose: To verify that Client is able to replay stored recording from Device by using GetReplayUri operation and RTSP commands to establish and then stop media stream with MPEG4 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetReplayUri operation and RTSP DESCRIBE, RTSP SETUP, RTSP PLAY and RTSP TEARDOWN commands present.

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

1. Client invokes GetReplayUri request message to retrieve URI of stored recording.
2. Device responds with code HTTP 200 OK and GetReplayUriResponse 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 MIME Type: MPEG4.
5. Client invokes RTSP SETUP request with transport parameter element to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes RTSP PLAY request 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.

Test Result:

NOTE: If Device RTSP DESCRIBE response message does not contain Media Type: "video" OR MIME Type: "MPEG4" inside SDP information then Test shall be deemed as "NOT DETECTED".

PASS -

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

- Client **GetReplayUri** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetReplayUri>" tag after the "<Body>" tag AND
  - [S3] "<GetReplayUri>" includes tag: "<RecordingToken>" with non-empty string value of specified record token AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<GetReplayUriResponse>" tag AND
  - [S6] "<GetReplayUriResponse>" includes tag: "<Uri>" with valid URI address AND
  - [S7] Client request introduces RTSP DESCRIBE command AND
- Client RTSP DESCRIBE request command has a proper hierarchy (see [RFC 2326]) AND
  - [S8] RTSP DESCRIBE includes: URI address obtained from GetReplayUriResponse AND
  - [S9] RTSP DESCRIBE includes: "RTSP/*" version identifier AND
  - [S10] RTSP DESCRIBE includes: "CSeq" identifier AND
  - [S11] Device response contains "RTSP/* 200 OK" AND
  - [S12] Device response SDP information contains Media Type: "video" and MIME Type: "MPEG4" AND
  - [S13] Client request introduces RTSP SETUP command AND
- Client RTSP SETUP request command has a proper hierarchy (see [RFC 2326]) AND
  - [S14] RTSP SETUP includes: URI address AND
  - [S15] RTSP SETUP includes: "RTSP/*" version identifier AND
  - [S16] RTSP SETUP includes: "CSeq" identifier AND
  - [S17] RTSP SETUP includes: "Transport" parameter AND
  - [S18] RTSP SETUP includes: "Require" parameter with "onvif-replay" value AND
  - [S19] Device response contains "RTSP/* 200 OK" AND
  - [S20] Client request introduces RTSP PLAY command AND
- Client RTSP PLAY request command has a proper hierarchy (see [RFC 2326]) AND
  - [S21] RTSP PLAY includes: URI address AND

- [S22] RTSP PLAY includes: "RTSP/*" version identifier AND
- [S23] RTSP PLAY includes: "CSeq" identifier AND
- [S24] RTSP PLAY includes: "Session" parameter AND
- [S25] RTSP PLAY includes: "Require" parameter with "onvif-replay" value AND
- [S26] Device response contains "RTSP/* 200 OK" AND
- [S27] Client request introduces RTSP TEARDOWN command AND

- Client RTSP TEARDOWN request command has a proper hierarchy (see [RFC 2326]) AND
  - [S28] RTSP TEARDOWN includes: URI address AND
  - [S29] RTSP TEARDOWN includes: "RTSP/*" version identifier AND
  - [S30] RTSP TEARDOWN includes: "CSeq" identifier AND
  - [S31] RTSP TEARDOWN includes: "Session" parameter AND
  - [S32] Device response contains "RTSP/* 200 OK" if it is detected.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_MPEG4ReplayRecording

6.5 H264 REPLAY RECORDING

Test Label: Replay Control - H264 Replay Recording

Test Case ID: REPLAYCONTROL-4

Profile G Normative Reference: Mandatory

Feature Under Test: Replay Control

Test Purpose: To verify that Client is able to replay stored recording from Device by using GetReplayUri operation and RTSP commands to establish and then stop media stream with H264 encoding type.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetReplayUri operation and RTSP DESCRIBE, RTSP SETUP, RTSP PLAY and RTSP TEARDOWN commands present.

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

1. Client invokes GetReplayUri request message to retrieve URI of stored recording.
2. Device responds with code HTTP 200 OK and GetReplayUriResponse 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 MIME Type: H264.
5. Client invokes RTSP SETUP request with transport parameter element to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes RTSP PLAY request 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.

Test Result:

NOTE: If Device RTSP DESCRIBE response message does not contain Media Type: "video" OR MIME Type: "H264" inside SDP information then Test shall be deemed as "NOT DETECTED".

PASS -

- Client **GetReplayUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetReplayUri** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetReplayUri>" tag after the "<Body>" tag AND
  - [S3] "<GetReplayUri>" includes tag: "<RecordingToken>" with non-empty string value of specified record token AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<GetReplayUriResponse>" tag AND

- [S6] "<GetReplayUriResponse>" includes tag: "<Uri>" with valid URI address AND
- [S7] Client request introduces RTSP DESCRIBE command AND

- Client RTSP DESCRIBE request command has a proper hierarchy (see [RFC 2326]) AND
  - [S8] RTSP DESCRIBE includes: URI address obtained from GetReplayUriResponse AND
  - [S9] RTSP DESCRIBE includes: "RTSP/*" version identifier AND
  - [S10] RTSP DESCRIBE includes: "CSeq" identifier AND
  - [S11] Device response contains "RTSP/* 200 OK" AND
  - [S12] Device response SDP information contains Media Type: "video" and MIME Type: "H264" AND
  - [S13] Client request introduces RTSP SETUP command AND
- Client RTSP SETUP request command has a proper hierarchy (see [RFC 2326]) AND
  - [S14] RTSP SETUP includes: URI address AND
  - [S15] RTSP SETUP includes: "RTSP/*" version identifier AND
  - [S16] RTSP SETUP includes: "CSeq" identifier AND
  - [S17] RTSP SETUP includes: "Transport" parameter AND
  - [S18] RTSP SETUP includes: "Require" parameter with "onvif-replay" value AND
  - [S19] Device response contains "RTSP/* 200 OK" AND
  - [S20] Client request introduces RTSP PLAY command AND
- Client RTSP PLAY request command has a proper hierarchy (see [RFC 2326]) AND
  - [S21] RTSP PLAY includes: URI address AND
  - [S22] RTSP PLAY includes: "RTSP/*" version identifier AND
  - [S23] RTSP PLAY includes: "CSeq" identifier AND
  - [S24] RTSP PLAY includes: "Session" parameter AND
  - [S25] RTSP PLAY includes: "Require" parameter with "onvif-replay" value AND
  - [S26] Device response contains "RTSP/* 200 OK" AND

- [S27] Client request introduces RTSP TEARDOWN command AND

- Client RTSP TEARDOWN request command has a proper hierarchy (see [RFC 2326]) AND
  - [S28] RTSP TEARDOWN includes: URI address AND
  - [S29] RTSP TEARDOWN includes: "RTSP/*" version identifier AND
  - [S30] RTSP TEARDOWN includes: "CSeq" identifier AND
  - [S31] RTSP TEARDOWN includes: "Session" parameter AND
  - [S32] Device response contains "RTSP/* 200 OK" if it is detected.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_H264ReplayRecording

6.6 REVERSE REPLAY

Test Label: Replay Control - Reverse Replay

Test Case ID: REPLAYCONTROL-5

Profile G Normative Reference: Optional

Feature Under Test: Replay Control

Test Purpose: To verify that Client is able to initiate a reverse playback of stored recording from Device by using the negative value of Scale header field in RTSP PLAY command.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with RTSP PLAY command with negative value of Scale header field.

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

1. Client invokes RTSP PLAY command with negative value of Scale header field to start reverse playback.
2. Device responds with code RTSP 200 OK.

Test Result:

PASS -

- [S1] Client request introduces RTSP PLAY command AND
- Client RTSP PLAY request command has a proper hierarchy (see [RFC 2326]) AND
  - [S2] RTSP PLAY includes: URI address AND
  - [S3] RTSP PLAY includes: "RTSP/*" version identifier AND
  - [S4] RTSP PLAY includes: "CSeq" identifier AND
  - [S5] RTSP PLAY includes: "Session" parameter AND
  - [S6] RTSP PLAY includes: "Require" parameter with "onvif-replay" value AND
  - [S7] RTSP PLAY includes: "Scale" parameter with any negative value (example: "-1.0", "-2", ...) AND
  - [S8] Device response contains "RTSP/* 200 OK".

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_ReverseReplay

6.7 RTSP SESSION TIMEOUT CONFIGURATION

Test Label: Replay Control - RTSP Session Timeout Configuration

Test Case ID: REPLAYCONTROL-6

Profile G Normative Reference: Conditional

Feature Under Test: Replay Control

Test Purpose: To verify that Client is able to change the RTSP session timeout configuration of the replay service using SetReplayConfiguration and GetReplayConfiguration operations.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with SetReplayConfiguration and GetReplayConfiguration operations present.

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

1. Client invokes SetReplayConfiguration request with a new duration value of SessionTimeout configuration.

2. Device responds with code HTTP 200 OK and SetReplayConfigurationResponse message.
3. Client invokes GetReplayConfiguration request to verify the current configuration of the replay service.
4. Device responds with code HTTP 200 OK and GetReplayConfigurationResponse message with current configuration listed.

Test Result:

PASS -

- Client **SetReplayConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetReplayConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<SetReplayConfiguration>" tag after the "<Body>" tag AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<SetReplayConfigurationResponse>" tag AND
- Client **GetReplayConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetReplayConfiguration** request in Test Procedure fulfills the following requirements:
  - [S6] Client request contains "<GetReplayConfiguration>" tag after the "<Body>" tag AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<GetReplayConfigurationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: ReplayControl_RTSPSessionTimeout

7 Recording Control – Dynamic Recording Configurations Test Cases

7.1 Feature Level Requirement:

Validated Feature: DynamicRecordingConfigurations

Profile G Requirement: Conditional

7.2 Expected Scenarios Under Test:

1. Client creates new Recording.
2. Client delete a Recording.
3. Client is considered as supporting Dynamic Recording Configurations if the following conditions are met depending on Device features:
   - Client is able to create a recording using the **CreateRecording** operation if Device supports DynamicRecordings feature AND
   - Client is able to delete a recording using the **DeleteRecording** operation if Device supports DynamicRecordings feature.
4. Client is considered as NOT supporting Dynamic Recording Configurations if ANY of the following is TRUE depending on Device features:
   - No Valid Device Response to **CreateRecording** request (except SOAP fault: **soapenv:Receiver/ter:Action/ter:MaxRecordings**) if Device supports DynamicRecordings feature OR
   - No Valid Device Response to **DeleteRecording** request (except SOAP fault: **soapenv:Receiver/ter:Action/ter:CannotDelete**) if Device supports DynamicRecordings feature.

7.3 CREATE A RECORDING

Test Label: Dynamic Recording Configurations - Create a Recording

Test Case ID: DYNAMICRECORDINGCONFIGURATION-1

Profile G Normative Reference: Conditional

Feature Under Test: Dynamic Recording Configurations

Test Purpose: To verify that Client is able to create a new recording on Device by using the **CreateRecording** operation.

Pre-Requisite:

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

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

1. Client invokes **CreateRecording** request message to create a new recording structure.
2. Device responds with code HTTP 200 OK and **CreateRecordingResponse** message or Device responds with **soapenv:Receiver/ter:Action/ter:MaxRecordings** SOAP fault.

Test Result:

PASS -

- Client **CreateRecording** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateRecording** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:CreateRecording** AND
- Device response on the **CreateRecording** request fulfills the following requirements:
  - [S2] IF it has HTTP 200 response code THEN **soapenv:Body** element has child element **trc:CreateRecordingResponse**
  - [S3] ELSE IF it has other than HTTP 200 response code THEN **soapenv:Body** element has child element **soapenv:Fault** with **soapenv:Receiver/ter:Action/ter:MaxRecordings** fault code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_CreateRecording

7.4 DELETE A RECORDING

Test Label: Dynamic Recording Configurations - Delete a Recording

Test Case ID: DYNAMICRECORDINGCONFIGURATION-2

Profile G Normative Reference: Conditional

Feature Under Test: Dynamic Recording Configurations

Test Purpose: To verify that Client is able to delete a recording on Device by using the **DeleteRecording** operation.

Pre-Requisite:

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

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

1. Client invokes **DeleteRecording** request message to delete a recording.
2. Device responds with code HTTP 200 OK and **DeleteRecordingResponse** message or Device responds with **soapenv:Receiver/ter:Action/ter:CannotDelete** SOAP fault.

Test Result:

PASS -

- Client **DeleteRecording** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteRecording** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:DeleteRecording** AND
  - [S2] **trc:DeleteRecording/trc:RecordingToken** element has non-empty string value of specific recording token AND
- Device response on the **DeleteRecording** request fulfills the following requirements:
  - [S3] IF it has HTTP 200 response code THEN **soapenv:Body** element has child element **trc:DeleteRecordingResponse**
  - [S4] ELSE IF it has other than HTTP 200 response code THEN **soapenv:Body** element has child element **soapenv:Fault** with **soapenv:Receiver/ter:Action/ter:CannotDelete** fault code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_DeleteRecording

8 Recording Control – Dynamic Track Configurations Test Cases

8.1 Feature Level Requirement:

Validated Feature: DynamicTracksConfigurations

Profile G Requirement: Conditional

8.2 Expected Scenarios Under Test:

1. Client creates new Track.
2. Client delete Track.
3. Client is considered as supporting Dynamic Tracks Configurations if the following conditions are met depending on Device features:
   - Client is able to create a track using the **CreateTrack** operation if Device supports DynamicTracks feature AND
   - Client is able to delete a track using the **DeleteTrack** operation if Device supports DynamicTracks feature.
4. Client is considered as NOT supporting Dynamic Tracks Configurations if ANY of the following is TRUE depending on Device features:
   - No Valid Device Response to **CreateTrack** request (except SOAP fault: **soapenv:Receiver/ter:Action/ter:MaxTracks**) if Device supports DynamicTracks feature OR
   - No Valid Device Response to **DeleteTrack** request (except SOAP fault: **soapenv:Receiver/ter:Action/ter:CannotDelete**) if Device supports DynamicTracks feature.

8.3 CREATE A TRACK

Test Label: Dynamic Tracks Configurations - Create a Track

Test Case ID: DYNAMICTRACKSCONFIGURATION-1

Profile G Normative Reference: Conditional

Feature Under Test: Dynamic Tracks Configurations

Test Purpose: To verify that Client is able to create a new track within a recording on Device by using the **CreateTrack** operation.

Pre-Requisite:

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

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

1. Client invokes **CreateTrack** request message with specified RecordingToken attribute to create a new track structure.
2. Device responds with code HTTP 200 OK and **CreateTrackResponse** message or Device responds with **soapenv:Receiver/ter:Action/ter:MaxTracks** SOAP fault.

Test Result:

PASS -

- Client **CreateTrack** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateTrack** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:CreateTrack** AND
  - [S2] **trc:CreateTrack/trc:RecordingToken** element has non-empty string value of specific recording token AND
  - [S3] **trc:CreateTrack/trc:TrackConfiguration/tt:TrackType** element value is equal to the strings "Video" OR "Audio" OR "Metadata" AND
- Device response on the **CreateTrack** request fulfills the following requirements:
  - [S4] IF it has HTTP 200 response code THEN **soapenv:Body** element has child element **trc:CreateTrackResponse**
  - [S5] ELSE IF it has other than HTTP 200 response code THEN **soapenv:Body** element has child element **soapenv:Fault** with **soapenv:Receiver/ter:Action/ter:MaxTracks** fault code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicTracksConfigurations_CreateTrack

8.4 DELETE A TRACK

Test Label: Dynamic Tracks Configurations - Delete a Track

Test Case ID: DYNAMICTRACKSCONFIGURATION-2

Profile G Normative Reference: Conditional

Feature Under Test: Dynamic Tracks Configurations

Test Purpose: To verify that Client is able to delete a track within a recoding on Device by using the **DeleteTrack** operation.

Pre-Requisite:

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

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

1. Client invokes **DeleteTrack** request message with specified RecordingToken attribute to delete a track.
2. Device responds with code HTTP 200 OK and **DeleteTrackResponse** message or Device responds with **soapenv:Receiver/ter:Action/ter:CannotDelete** SOAP fault.

Test Result:

PASS -

- Client **DeleteTrack** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteTrack** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:DeleteTrack** AND
  - [S2] **trc:DeleteTrack/trc:RecordingToken** element has non-empty string value of specific recording token AND
  - [S3] **trc:DeleteTrack/trc:TrackToken** element has non-empty string value of specific track token AND
- Device response on the **DeleteTrack** request fulfills the following requirements:

- [S4] IF it has HTTP 200 response code THEN **soapenv:Body** element has child element **trc:DeleteTrackResponse**
- [S5] ELSE IF it has other than HTTP 200 response code THEN **soapenv:Body** element has child element **soapenv:Fault** with **soapenv:Receiver/ter:Action/ter:CannotDelete** fault code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicTrackConfigurations_DeleteTrack

9 Recording Control Test Cases

9.1 Feature Level Requirement:

Validated Feature: RecordingControl

Profile G Requirement: Conditional

9.2 Expected Scenarios Under Test:

1. Client retrieves a list of recordings using **GetRecordings** operation.
2. Client retrieves a list of recording jobs using **GetRecordingJobs** operation.
3. Client managing recording jobs on a device using **CreateRecordingJob** and **DeleteRecordingJob** operations.
4. Client managing the mode of a recording job on a device using **SetRecordingJobMode** operation.
5. Client may get state of a recording job on a device using **GetRecordingJobState** operation.
6. Client retrieves notifications of the change in a recording job's state using Pull Point event mechanism.
7. When Device and Client support dynamic update of track/recording content capability:
   - Client retrieves notifications of the change in a recording's content using Pull Point event mechanism.
8. Client is considered as supporting Recording Control if the following conditions are met:
   - Client is able to retrieve a list of recordings using the **GetRecordings** operation AND
   - Client is able to retrieve recording jobs using the **GetRecordingJobs** operation AND
   - Client is able to manage recording jobs on a device using the **CreateRecordingJob** and the **DeleteRecordingJob** operations AND
   - Client is able to manage the mode of a recording job on a device using the **SetRecordingJobMode** operation AND
   - Client supports EventHandling_Pullpoint feature AND
   - Client is able to retrieve tns1:RecordingConfig/JobState notification about change of recording job's state AND

- Client is able to retrieve tns1:RecordingConfig/DeleteTrackData notification about change of recording's content if Device supports RecordingConfigDeleteTrackDataEvent feature.

9. Client is considered as NOT supporting Recording Control if ANY of the following is TRUE:
   - No Valid Device Response to **GetRecordings** request OR
   - No Valid Device Response to **GetRecordingJobs** request OR
   - No Valid Device Response to **CreateRecordingJob** request (except SOAP fault: EITHER **soapenv:Receiver/ter:Action/ter:MaxRecordingJobs** OR **env:Receiver/ter:Action/ter:MaxReceivers**) OR
   - No Valid Device Response to **DeleteRecordingJob** request OR
   - No Valid Device Response to **GetRecordingJobState** request if detected OR
   - No Valid Device Response to **SetRecordingJobMode** request OR
   - Client does not support EventHandling_Pullpoint feature OR
   - Client unable to retrieve tns1:RecordingConfig/JobState notification about change of recording job's state OR
   - Client unable to retrieve tns1:RecordingConfig/DeleteTrackData notification about change of recording's content if Device supports RecordingConfigDeleteTrackDataEvent feature.

9.3 GET RECORDINGS

Test Label: Recording Control - Get Recordings

Test Case ID: RECORDINGCONTROL-1

Profile G Normative Reference: Conditional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to receive a list of recordings from Device using the **GetRecordings** operation.

Pre-Requisite:

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

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

1. Client invokes **GetRecordings** request message.
2. Device responds with code HTTP 200 OK and **GetRecordingsResponse** message.

Test Result:

PASS -

- Client **GetRecordings** request message is valid according to XML Schemas listed in Namespaces AND
  - [S1] Client request contains "<GetRecordings>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetRecordingsResponse>" tag

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_GetRecordings

9.4 GET RECORDING JOBS

Test Label: Recording Control - Get Recording Jobs

Test Case ID: RECORDINGCONTROL-2

Profile G Normative Reference: Conditional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to receive a list of recording jobs from Device using the **GetRecordingJobs** operation.

Pre-Requisite:

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

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

1. Client invokes **GetRecordingJobs** request message.

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

Test Result:

PASS -

- Client **GetRecordingJobs** request message is valid according to XML Schemas listed in Namespaces AND
  - [S1] Client request contains "<GetRecordingJobs>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetRecordingJobsResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_GetRecordingJobs

9.5 GET RECORDING JOB STATE

Test Label: Recording Control - Get Recording Job State

Test Case ID: RECORDINGCONTROL-3

Profile G Normative Reference: Optional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to receive a specific recording jobs state from Device using the **GetRecordingJobState** operation.

Pre-Requisite:

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

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

1. Client invokes **GetRecordingJobState** request message.
2. Device responds with code HTTP 200 OK and **GetRecordingJobStateResponse** message.

Test Result:

PASS -

- Client **GetRecordingJobState** request message is valid according to XML Schemas listed in Namespaces AND
  - [S1] Client request contains "<GetRecordingJobState>" tag after the "<Body>" tag AND
  - [S2] "<JobToken>" tag in request has non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<GetRecordingJobStateResponse>" tag

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_GetRecordingJobState

9.6 MODIFY RECORDING JOB MODE

Test Label: Recording Control - Set Recording Job Mode

Test Case ID: RECORDINGCONTROL-4

Profile G Normative Reference: Conditional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to manage the mode of a recording job on Device using the **SetRecordingJobMode** operation.

Pre-Requisite:

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

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

1. Client invokes **SetRecordingJobMode** request message.
2. Device responds with code HTTP 200 OK and **SetRecordingJobModeResponse** message.

Test Result:

PASS -

- Client **SetRecordingJobMode** request message is valid according to XML Schemas listed in Namespaces AND
  - [S1] Client request contains "<SetRecordingJobMode>" tag after the "<Body>" tag AND
  - [S2] "<JobToken>" tag in request has non-empty string value of specific token AND
  - [S3] "<Mode>" tag in request has value equals EITHER ("Active" OR "Idle") AND
  - [S4] Device response contains "HTTP/* 200 OK" AND
  - [S5] Device response contains "<SetRecordingJobModeResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_SetRecordingJobMode

9.7 CREATE A RECORDING JOB

Test Label: Recording Control - Create Recording Job

Test Case ID: RECORDINGCONTROL-5

Profile G Normative Reference: Conditional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to create a new recording job on Device by using the **CreateRecordingJob** operation.

Pre-Requisite:

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

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

1. Client invokes **CreateRecordingJob** request message.
2. Device responds with code HTTP 200 OK and **CreateRecordingJobResponse** message or Device responds with **soapenv:Receiver/ter:Action/ter:MaxRecordingJobs** SOAP fault OR **soapenv:Receiver/ter:Action/ter:MaxReceivers** SOAP fault.

Test Result:

PASS -

- Client **CreateRecordingJob** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateRecordingJob** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:CreateRecordingJob** AND
  - [S2] **trc:CreateRecordingJob/trc:JobConfiguration/tt:RecordingToken** element has non-empty string value of specific recording token AND
  - [S3] **trc:CreateRecordingJob/trc:JobConfiguration/tt:Mode** element has value is equal to EITHER "Active" OR "Idle" AND
  - [S4] **trc:CreateRecordingJob/trc:JobConfiguration/tt:Priority** element has a non-negative value AND
- Device response on the **CreateRecordingJob** request fulfills the following requirements:
  - [S5] IF it has HTTP 200 response code THEN **soapenv:Body** element has child element **trc:CreateRecordingJob**
  - [S6] ELSE IF it has other than HTTP 200 response code THEN **soapenv:Body** element has child element **soapenv:Fault** with **soapenv:Receiver/ter:Action/ter:MaxRecordingJobs** fault code OR **soapenv:Receiver/ter:Action/ter:MaxReceivers** fault code.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_CreateRecordingJob

9.8 DELETE A RECORDING JOB

Test Label: Recording Control - Delete Recording Job

Test Case ID: RECORDINGCONTROL-6

Profile G Normative Reference: Conditional

Feature Under Test: Recording Control

Test Purpose: To verify that Client is able to delete a recording job on Device by using the **DeleteRecordingJob** operation.

Pre-Requisite:

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

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

1. Client invokes **DeleteRecordingJob** request message to delete a recording job.
2. Device responds with code HTTP 200 OK and **DeleteRecordingJobResponse** message.

Test Result:

PASS -

- Client **DeleteRecordingJob** request message is valid according to XML Schemas listed in Namespaces AND
  - [S1] Client request contains "<DeleteRecordingJob>" tag after the "<Body>" tag AND
  - [S2] "<JobToken>" tag in request has non-empty string value of specific recording job token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<DeleteRecordingJobResponse>" tag.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: RecordingControl_DeleteRecordingJob

10 Recording Configuration Test Cases

10.1 Feature Level Requirement:

Validated Feature: recording_configuration

Profile G Requirement: Conditional

10.2 Expected Scenarios Under Test:

1. Client retrieves configuration of recording using **GetRecordingConfiguration** operation.
2. Client manages a recording's configuration using **SetRecordingConfiguration** operation.
3. Client retrieves configuration of recording's job using **GetRecordingJobConfiguration** operation.
4. Client manages a recording job's configuration using **SetRecordingJobConfiguration** operation.
5. Client retrieves notifications of the change in a recording's configuration using Pull Point event mechanism, if Device supports recording configuration change event capability.
6. Client retrieves notifications of the change in a recording job's configuration using Pull Point event mechanism, if Device supports recording job's configuration change event capability.
7. Client is considered as supporting Recording Configuration if the following conditions are met:
   - Client is able to retrieve configuration of recording using **GetRecordingConfiguration** operation AND
   - Client is able to manage a recording's configuration using **SetRecordingConfiguration** operation AND
   - Client is able to retrieve configuration of recording's job using **GetRecordingJobConfiguration** operation AND
   - Client is able to manage a recording job's configuration using **SetRecordingJobConfiguration** operation AND
   - Client supports EventHandling_Pullpoint feature AND

- Client is able to retrieve tns1:RecordingConfig/RecordingConfiguration notification about change in a recording's configuration if Device supports RecordingConfigRecordingConfigurationEvent feature AND
- Client is able to retrieve tns1:RecordingConfig/RecordingJobConfiguration notification about change in a recording job's configuration if Device supports RecordingConfigRecordingJobConfigurationEvent feature.

8. Client is considered as NOT supporting Recording Configuration if ANY of the following is TRUE:
   - No Valid Device Response to **GetRecordingConfiguration** request OR
   - No Valid Device Response to **SetRecordingConfiguration** request OR
   - No Valid Device Response to **GetRecordingJobConfiguration** request OR
   - No Valid Device Response to **SetRecordingJobConfiguration** request OR
   - Client does not support EventHandling_Pullpoint feature if Device supports RecordingConfigRecordingConfigurationEvent feature or RecordingConfigRecordingJobConfigurationEvent feature OR
   - Client unable to retrieve tns1:RecordingConfig/RecordingConfiguration notification about change of recording's configuration if Device supports RecordingConfigRecordingConfigurationEvent feature OR
   - Client unable to retrieve tns1:RecordingConfig/RecordingJobConfiguration notification about change in a recording job's configuration if Device supports RecordingConfigRecordingJobConfigurationEvent feature.

10.3 GET RECORDING CONFIGURATION

Test Label: Recording Configuration - Get Recording Configuration

Test Case ID: RECORDINGCONFIGURATION-1

Profile G Normative Reference: Conditional

Feature Under Test: Recording Configuration

Test Purpose: To verify that Client is able to receive configuration of a recording from Device using the **GetRecordingConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **GetRecordingConfiguration** request message.
2. Device responds with code HTTP 200 OK and **GetRecordingConfigurationResponse** message.

Test Result:

PASS -

- Client **GetRecordingConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **GetRecordingConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:GetRecordingConfiguration** AND
  - [S2] **trc:RecordingToken** element has non-empty value AND
- Device response on the **GetRecordingConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trc:GetRecordingConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: recording_configuration.get_recording_configuration

10.4 SET RECORDING CONFIGURATION

Test Label: Recording Configuration - Set Recording Configuration

Test Case ID: RECORDINGCONFIGURATION-2

Profile G Normative Reference: Conditional

Feature Under Test: Recording Configuration

Test Purpose: To verify that Client is able to manages a recording's configuration using the **SetRecordingConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **SetRecordingConfiguration** request message.
2. Device responds with code HTTP 200 OK and **SetRecordingConfigurationResponse** message.

Test Result:

PASS -

- Client **SetRecordingConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **SetRecordingConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:SetRecordingConfiguration** AND
  - [S2] **trc:RecordingToken** element has non-empty value AND
- Device response on the **SetRecordingConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trc:SetRecordingConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: recording_configuration.set_recording_configuration

10.5 GET RECORDING JOB CONFIGURATION

Test Label: Recording Configuration - Get Recording Job Configuration

Test Case ID: RECORDINGCONFIGURATION-3

Profile G Normative Reference: Conditional

Feature Under Test: Recording Configuration

Test Purpose: To verify that Client is able to receive configuration of a recording job from Device using the **GetRecordingJobConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **GetRecordingJobConfiguration** request message.
2. Device responds with code HTTP 200 OK and **GetRecordingJobConfigurationResponse** message.

Test Result:

PASS -

- Client **GetRecordingJobConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **GetRecordingJobConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:GetRecordingJobConfiguration** AND
  - [S2] **trc:JobToken** element has non-empty value AND
- Device response on the **GetRecordingJobConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trc:GetRecordingJobConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: recording_configuration.get_recording_job_configuration

10.6 SET RECORDING JOB CONFIGURATION

Test Label: Recording Configuration - Set Recording Job Configuration

Test Case ID: RECORDINGCONFIGURATION-4

Profile G Normative Reference: Conditional

Feature Under Test: Recording Configuration

Test Purpose: To verify that Client is able to manages a recording job configuration using the **SetRecordingJobConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **SetRecordingJobConfiguration** request message.
2. Device responds with code HTTP 200 OK and **SetRecordingJobConfigurationResponse** message.

Test Result:

PASS -

- Client **SetRecordingJobConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **SetRecordingJobConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:SetRecordingJobConfiguration** AND
  - [S2] **trc:JobToken** element has non-empty value AND
  - [S3] **trc:JobConfiguration/tt:RecordingToken** element has non-empty value AND
  - [S4] **trc:JobConfiguration/tt:Mode** element value equals to EITHER "Active" OR "Idle" AND
  - [S5] **trc:JobConfiguration/tt:Priority** element has a non-negative value AND
  - For each **trc:JobConfiguration/tt:Source** element:

- If does not contain **tt:SourceToken** element then it fulfills the following requirements (else skip the check):
  - [S6] It contains **tt:AutoCreateReceiver** element with value equals to true AND
- If contains **tt:SourceToken** element then it fulfills the following requirements (else skip the checks):
  - [S7] **tt:SourceToken/tt:Token** element has non-empty string value of specific token AND
  - If it contains **tt:SourceToken/@Type** attribute equal to **"http://www.onvif.org/ver10/schema/Profile"** then it fulfills the following requirements (else skip the check):
    - [S8] It does not contain **tt:AutoCreateReceiver** element AND

- Device response on the **SetRecordingJobConfiguration** request fulfills the following requirements:
  - [S9] It has HTTP 200 response code AND
  - [S10] **soapenv:Body** element has child element **trc:SetRecordingJobConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: recording_configuration.set_recording_job_configuration

11 Track Configuration Test Cases

11.1 Feature Level Requirement:

Validated Feature: track_configuration

Profile G Requirement: Conditional

11.2 Expected Scenarios Under Test:

1. Client retrieves configuration of track using **GetTrackConfiguration** operation.
2. Client manages a track's configuration using **SetTrackConfiguration** operation.
3. Client retrieves notifications of the change in a track's configuration using Pull Point event mechanism if Device supports track configuration change event capability.
4. Client is considered as supporting Track Configuration if the following conditions are met:
   - Client is able to retrieve configuration of track using **GetTrackConfiguration** operation AND
   - Client is able to manage a track's configuration using **SetTrackConfiguration** operation AND
   - Client supports EventHandling_Pullpoint feature AND
   - Client is able to retrieve tns1:RecordingConfig/TrackConfiguration notification about change in a track's configuration if Device supports RecordingConfigTrackConfigurationEvent feature.
5. Client is considered as NOT supporting Track Configuration if ANY of the following is TRUE:
   - No Valid Device Response to **GetTrackConfiguration** request OR
   - No Valid Device Response to **SetTrackConfiguration** request OR
   - Client does not support EventHandling_Pullpoint feature if Device supports RecordingConfigTrackConfigurationEvent feature OR
   - Client unable to retrieve tns1:RecordingConfig/TrackConfiguration notification about change of track's configuration if Device supports RecordingConfigTrackConfigurationEvent feature.

11.3 GET TRACK CONFIGURATION

Test Label: Track Configuration - Get Track Configuration

Test Case ID: TRACKCONFIGURATION-1

Profile G Normative Reference: Conditional

Feature Under Test: Track Configuration

Test Purpose: To verify that Client is able to receive configuration of a track from Device using the **GetTrackConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **GetTrackConfiguration** request message.
2. Device responds with code HTTP 200 OK and **GetTrackConfigurationResponse** message.

Test Result:

PASS -

- Client **GetTrackConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **GetTrackConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:GetTrackConfiguration** AND
  - [S2] **trc:RecordingToken** element has non-empty value AND
  - [S3] **trc:TrackToken** element has non-empty value AND
- Device response on the **GetTrackConfiguration** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **trc:GetTrackConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: track_configuration.get_track_configuration

11.4 SET TRACK CONFIGURATION

Test Label: Recording Configuration - Set Track Configuration

Test Case ID: TRACKCONFIGURATION-2

Profile G Normative Reference: Conditional

Feature Under Test: Track Configuration

Test Purpose: To verify that Client is able to manages a track's configuration using the **SetTrackConfiguration** operation.

Pre-Requisite:

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

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

1. Client invokes **SetTrackConfiguration** request message.
2. Device responds with code HTTP 200 OK and **SetTrackConfigurationResponse** message.

Test Result:

PASS -

- Client **SetTrackConfiguration** request message is valid according to XML Schemas listed in Namespaces AND
- Client **SetTrackConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trc:SetTrackConfiguration** AND
  - [S2] **trc:RecordingToken** element has non-empty value AND
  - [S3] **trc:TrackToken** element has non-empty value AND
  - [S4] **trc:TrackConfiguration/tt:TrackType** element value equals EITHER "Video" OR "Audio" OR "Metadata" AND
- Device response on the **SetTrackConfiguration** request fulfills the following requirements:

- [S5] It has HTTP 200 response code AND
- [S6] **soapenv:Body** element has child element **trc:SetTrackConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: track_configuration.set_track_configuration

12 Recording Control – Using a Receiver as Source Test Cases

12.1 Feature Level Requirement:

Validated Feature: using_receiver_as_source

Profile G Requirement: Conditional

12.2 Expected Scenarios Under Test:

1. Client retrieves a list of receivers from Device using the **GetReceivers** operation.
2. Client retrieves receiver configuration from Device using the **GetReceiver** operation or **GetReceivers** operation.
3. Client creates new receiver using the **CreateReceiver** operation.
4. Client deletes a receiver using the **DeleteReceiver** operation.
5. Client configures a receiver using the **ConfigureReceiver** operation.
6. Client may get state of a recording job on a device using **GetReceiverState** operation.
7. Client retrieves the state of a receiver using subscribes to device messages using **CreatePullPointSubscription** operation to get receiver's state changed notifications.
8. Client retrieves the connection state of a receiver using the subscribes to device messages using **CreatePullPointSubscription** operation to get receiver's state changed notifications OR subscribes to device messages using **CreatePullPointSubscription** operation to get connection failed notifications.
9. Client sets the mode of a receiver using the **SetReceiverMode** operation.
10. Client uses Pull Point event mechanism to retrieve notification events from Device.
11. Client is considered as supporting Using a Receiver as Source if the following conditions are met:
    - Client supports EventHandling_PullPoint feature to use Pull Point event mechanism to retrieve notification events from Device AND
    - Client is able to retrieve list of receivers using **GetReceivers** operation AND
    - Client is able to retrieve receiver configuration using **GetReceiver** operation OR **GetReceivers** operation AND

- Client is able to create new receiver using **CreateReceiver** operation AND
- Client is able to delete receiver using **DeleteReceiver** operation AND
- Client is able to configure receiver using **ConfigureReceiver** operation AND
- Client is able to retrieve **tns1:Receiver/ChangeState** notification about change in a receiver's state AND
- Client is able to retrieve **tns1:Receiver/ChangeState** notification OR **tns1:Receiver/ConnectionFailed** notification AND
- Client is able to set receiver mode using **SetReceiverMode** operation.

12. Client is considered as NOT supporting Using a Receiver as Source if ANY of the following is TRUE:
    - Client does not support EventHandling_PullPoint feature OR
    - No Valid Device Response to **GetReceivers** request OR
    - No Valid Device Response to **GetReceiver** request if detected OR
    - No Valid Device Response to **CreateReceiver** request OR
    - No Valid Device Response to **DeleteReceiver** request OR
    - No Valid Device Response to **ConfigureReceiver** request OR
    - No Valid Device Response to **SetReceiverMode** request OR
    - No Valid Device Response to **GetReceiverState** request if detected OR
    - Client unable to retrieve **tns1:Receiver/ChangeState** notification about change in a receiver's state AND
    - Client unable to retrieve **tns1:Receiver/ChangeState** notification AND **tns1:Receiver/ConnectionFailed** notification AND

12.3 GET RECEIVERS

Test Label: Recording Control – Using a Receiver as Source - Get Receivers

Test Case ID: RECEIVER-1

Profile G Normative Reference: Conditional

Feature Under Test: Get Receivers

Test Purpose: To verify that list of receivers from Device is received by Client using the **GetReceivers** operation.

Pre-Requisite:

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

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

1. Client invokes **GetReceivers** request message to retrieve list of receivers from the Device.
2. Device responds with code HTTP 200 OK and **GetReceiversResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.get_recievers

12.4 GET RECEIVER

Test Label: Recording Control – Using a Receiver as Source - Get Receiver

Test Case ID: RECEIVER-2

Profile G Normative Reference: Conditional

Feature Under Test: Get Receiver

Test Purpose: To verify that receiver configuration from Device is received by Client using the **GetReceiver** operation.

Pre-Requisite:

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

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

1. Client invokes **GetReceiver** request message to retrieve receiver configuration from the Device.
2. Device responds with code HTTP 200 OK and **GetReceiverResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.get_reciever

12.5 CREATE RECEIVER

Test Label: Recording Control – Using a Receiver as Source - Create Receiver

Test Case ID: RECEIVER-3

Profile G Normative Reference: Conditional

Feature Under Test: Create Receiver

Test Purpose: To verify that receiver is created on Devicve by Client using the **CreateReceiver** operation.

Pre-Requisite:

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

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

1. Client invokes **CreateReceiver** request message to create receiver on the Device.
2. Device responds with code HTTP 200 OK and **CreateReceiverResponse** message.

Test Result:

PASS -

- Client **CreateReceiver** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateReceiver** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trv:CreateReceiver** AND
  - [S2] **trv:Configuration/tt:Mode** element value is not equal **"Unknown"** AND
- Device response on the **CreateReceiver** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trv:CreateReceiverResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.create_reciever

12.6 DELETE RECEIVER

Test Label: Recording Control – Using a Receiver as Source - Delete Receiver

Test Case ID: RECEIVER-4

Profile G Normative Reference: Conditional

Feature Under Test: Delete Receiver

Test Purpose: To verify that receiver is deleted on Devicve by Client using the **DeleteReceiver** operation.

Pre-Requisite:

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

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

1. Client invokes **DeleteReceiver** request message to delete receiver from the Device.
2. Device responds with code HTTP 200 OK and **DeleteReceiverResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.delete_reciever

12.7 CONFIGURE RECEIVER

Test Label: Recording Control – Using a Receiver as Source - Configure Receiver

Test Case ID: RECEIVER-5

Profile G Normative Reference: Conditional

Feature Under Test: Configure Receiver

Test Purpose: To verify that receiver is configured on Devicve by Client using the **ConfigureReceiver** operation.

Pre-Requisite:

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

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

1. Client invokes **ConfigureReceiver** request message to configure receiver on the Device.
2. Device responds with code HTTP 200 OK and **ConfigureReceiverResponse** message.

Test Result:

PASS -

- Client **ConfigureReceiver** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **ConfigureReceiver** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trv:ConfigureReceiver** AND
  - [S2] **trv:Configuration\tt:Mode** element value is not equal **"Unknown"** AND
- Device response on the **ConfigureReceiver** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trv:ConfigureReceiverResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.configure_reciever

12.8 GET RECEIVER STATE

Test Label: Recording Control – Using a Receiver as Source - Get Receiver State

Test Case ID: RECEIVER-6

Profile G Normative Reference: Optional

Feature Under Test: Get Receiver State

Test Purpose: To verify that receiver state from Device is received by Client using the **GetReceiverState** operation.

Pre-Requisite:

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

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

1. Client invokes **GetReceiverState** request message to request receiver state from the Device.
2. Device responds with code HTTP 200 OK and **GetReceiverStateResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.get_receiver_state

12.9 SET RECEIVER MODE

Test Label: Recording Control – Using a Receiver as Source - Set Receiver Mode

Test Case ID: RECEIVER-7

Profile G Normative Reference: Conditional

Feature Under Test: Set Receiver Mode

Test Purpose: To verify that receiver mode is changed by Client on Device using the **SetReceiverMode** operation.

Pre-Requisite:

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

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

1. Client invokes **SetReceiverMode** request message to change receiver mode on the Device.
2. Device responds with code HTTP 200 OK and **SetReceiverModeResponse** message.

Test Result:

PASS -

- Client **SetReceiverMode** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetReceiverMode** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **trv:SetReceiverMode** AND
  - [S2] **trv:Mode** element value is not equal **"Unknown"** AND
- Device response on the **SetReceiverMode** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **trv:SetReceiverModeResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: using_receiver_as_source.set_receiver_mode