ONVIF™

Imaging Client Test Specification

Version 16.07

July 2016

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

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

REVISION HISTORY

Vers.	Date	Description
16.01	December 18, 2016	Focus Control Test Cases was splitted on two parts (Focus Move Capabilities Test Cases and Focus Control Test Cases) to handle Device capabilities issue.
16.01	December 18, 2016	Get Imaging Capabilities Test Cases feature definition was updated
15.10	October 20, 2016	• Initial version: • General parts added • Get Imaging Capabilities Test Cases added • Video Sources List Test Cases added • Get Imaging Settings Test Cases added • Imaging Settings Configuration Test Cases added • Focus Control Test Cases added

Table of Contents

1 Introduction 6

1.1 Scope 6

1.2 Get Imaging Capabilities 7

1.3 Get Imaging Settings 7

1.4 Imaging Settings Configuration 7

1.5 Focus Move Capabilities 7

1.6 Focus Control 7

2 Normative references 8

3 Terms and Definitions 9

3.1 Conventions 9

3.2 Definitions 9

3.3 Abbreviations 9

3.4 Namespaces 10

4 Test Overview 11

4.1 Test Setup 11

4.2 Prerequisites 11

5 Get Imaging Capabilities Test Cases 12

5.1 Expected Scenarios Under Test: 12

5.2 GET CAPABILITIES 12

5.3 GET SERVICE CAPABILITIES 14

6 Get Imaging Settings Test Cases 16

6.1 Expected Scenarios Under Test: 16

6.2 GET IMAGING SETTINGS 16

7 Imaging Settings Configuration Test Cases 18

7.1 Expected Scenarios Under Test: 18

7.2 GET OPTIONS 18

7.3 SET IMAGING SETTINGS 19

8 Focus Move Capabilities Test Cases 29

8.1 Expected Scenarios Under Test: 29

8.2 GET FOCUS MOVE OPTIONS 29

9 Focus Control Test Cases 31

9.1 Expected Scenarios Under Test: 31

9.2 ABSOLUTE FOCUS MOVE 31

9.3 RELATIVE FOCUS MOVE 33

9.4 CONTINUOUS FOCUS MOVE 35

9.5 STOP 37

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 [http://www.onvif.org/Documents/Specifications.aspx]).

This particular document defines test cases required for testing Imaging Service features of a Client application e.g. Get Imaging Capabilities, Video Sources List, Get Imaging Settings, Imaging Settings Configuration, Focus Control. 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 Imaging Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Imaging Service 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 Imaging Service features according to ONVIF Imaging Service Specification.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Imaging Service features.

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

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

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

1.2 Get Imaging Capabilities

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

1.3 Get Imaging Settings

Get Imaging Settings section specifies Client ability to request imaging settings from Device.

1.4 Imaging Settings Configuration

Imaging Settings Configuration section specifies Client ability to change imaging settings on Device.

1.5 Focus Move Capabilities

Focus Move Capabilities section specifies Client ability to retrieve focus move capabilities from Device.

1.6 Focus Control

Focus Control section specifies Client ability to control focus on Device.

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 Imaging Specification:

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

- ISO/IEC Directives, Part 2:

  http://www.iso.org/directives

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

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

- 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/]

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

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.

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.

Imaging Service Services for exposure time, gain and white balance parameters among others.

Image Stabilization Functionality used to avoid blurring of images due to movement of the device or its objects.

Tone Compensation Functionality used to make the image with dark or bright areas to be more visible.

Defogging Functionality used to make the image more detailed in presence of fog.

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.

URI Uniform Resource Identifier.

WSDL Web Services Description Language.

XML eXtensible Markup Language.

3.4 Namespaces

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

Table 3.1. Defined namespaces in this specification

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
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
trt	http://www.onvif.org/ver10/media/wsdl	The namespace for the WSDL media service
timg	http://www.onvif.org/ver20/imaging/wsd	The namespace for the WSDL imaging service

4 Test Overview

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

An ONVIF client with Imaging features support can provide image settings configuration and focus control.

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 Test Setup

Collect Network Traces files required by the test cases.

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

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

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

4.2 Prerequisites

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

The Device shall be configured with an IPv4 address.

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

5 Get Imaging Capabilities Test Cases

Validated Feature: get_imaging_capabilities

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

5.1 Expected Scenarios Under Test:

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

5.2 GET CAPABILITIES

Test Label: Get Imaging Capabilities - Get Capabilities

Test Case ID: GETIMAGINGCAPABILITIES-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Capabilities

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

Pre-Requisite:

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

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

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

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_imaging_capabilities.get_capabilities

5.3 GET SERVICE CAPABILITIES

Test Label: Get Imaging Capabilities - Get Service Capabilities

Test Case ID: GETIMAGINGCAPABILITIES-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Services

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

Pre-Requisite:

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

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

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

Test Result:

PASS -

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

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_imaging_capabilities.get_service_capabilities

6 Get Imaging Settings Test Cases

Validated Feature: get_imaging_settings

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

6.1 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a current imaging settings.
2. Client is considered as supporting Get Imaging Settings if the following conditions are met:
    - Client is able to retrieve a current imaging settings using **GetImagingSettings** operation.
3. Client is considered as NOT supporting Get Imaging Settings if ANY of the following is TRUE:
    - No valid responses for **GetImagingSettings** request.

6.2 GET IMAGING SETTINGS

Test Label: Get Imaging Settings - Get Imaging Settings

Test Case ID: GETIMAGINGSETTINGS-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Imaging Settings

Test Purpose: To verify that imaging settings for Device is received by Client using the **GetImagingSettings** operation.

Pre-Requisite:

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

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

1. Client invokes **GetImagingSettings** request message to retrieve imaging settings for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetImagingSettingsResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_imaging_settings.get_imaging_settings

7 Imaging Settings Configuration Test Cases

Validated Feature: set_imaging_settings

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

7.1 Expected Scenarios Under Test:

1. Client connects to Device to change imaging settings.
2. Client is considered as supporting Imaging Settings Configuration if the following conditions are met:
   - Client is able to retrieve a imaging options using **GetOptions** operation AND
   - Client is able to change a imaging settings using **SetImagingSettings** operation.
3. Client is considered as NOT supporting Imaging Settings Configuration if ANY of the following is TRUE:
   - No valid responses for **GetOptions** request OR
   - No valid responses for **SetImagingSettings** request OR
   - **SetImagingSettings** request contains settings which does not correspong to **GetOptionsResponse** message for the same video source token.

7.2 GET OPTIONS

Test Label: Get Imaging Settings - Get Options

Test Case ID: SETIMAGINGSETTINGS-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Options

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

Pre-Requisite:

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

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

1. Client invokes **GetOptions** request message to retrieve imaging options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetOptionsResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_imaging_settings.get_options

7.3 SET IMAGING SETTINGS

Test Label: Set Imaging Settings - Set Imaging Settings

Test Case ID: SETIMAGINGSETTINGS-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Set Imaging Settings

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

Pre-Requisite:

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

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

1. Client invokes **GetOptions** request message to retrieve imaging options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetOptionsResponse** message.
3. Client invokes **SetImagingSettings** request message to change imaging settings for specified video source which are correspond to the resieved options on the Device.
4. Device responses with code HTTP 200 OK and **SetImagingSettingsResponse** message.

Test Result:

PASS -

- Client **SetImagingSettings** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetImagingSettings** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **timg:SetImagingSettings** AND
- Device response on the **SetImagingSettings** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **timg:SetImagingSettingsResponse** AND
- There is a Client **GetOptions** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **SetImagingSettings** request AND
  - [S5] It invoked before the Client **SetImagingSettings** request AND
  - [S6] **timg:VideoSourceToken** element value is equal to **timg:VideoSourceToken** element from the **SetImagingSettings** request AND

- Device response on the **GetOptions** request fulfills the following requirements:
  - [S7] It has HTTP 200 response code AND

- Settings from the **SetImagingSettings** request corresponds options recieved in the **GetOptionsResponse** message:
  - [S8] IF the **SetImagingSettings** request contains **tt:BacklightCompensation** element THEN:
    - The **GetOptionsResponse** message contains **tt:BacklightCompensation** element AND
    - The **GetOptionsResponse** message contains **tt:BacklightCompensation/tt:Mode** element value equal to **tt:BacklightCompensation/tt:Mode** element value from the **SetImagingSettings** request AND
  - [S9] IF the **SetImagingSettings** request contains **tt:BacklightCompensation/tt:Level** element THEN:
    - The **GetOptionsResponse** message contains **tt:BacklightCompensation/tt:Level** element AND
    - **tt:BacklightCompensation/tt:Level** element value from the **SetImagingSettings** request is less or equal to **tt:BacklightCompensation/tt:Level/tt:Max** from the the **GetOptionsResponse** message AND
    - **tt:BacklightCompensation/tt:Level** element value from the **SetImagingSettings** request is greater or equal to **tt:BacklightCompensation/tt:Level/tt:Min** from the the **GetOptionsResponse** message AND
  - [S10] IF the **SetImagingSettings** request contains **tt:Brightness** element THEN:
    - The **GetOptionsResponse** message contains **tt:Brightness** element AND
    - **tt:Brightness** element value from the **SetImagingSettings** request is less or equal to **tt:Brightness/tt:Max** from the the **GetOptionsResponse** message AND
    - **tt:Brightness** element value from the **SetImagingSettings** request is greater or equal to **tt:Brightness/tt:Min** from the the **GetOptionsResponse** message AND
  - [S11] IF the **SetImagingSettings** request contains **tt:ColorSaturation** element THEN:
    - The **GetOptionsResponse** message contains **tt:ColorSaturation** element AND
    - **tt:ColorSaturation** element value from the **SetImagingSettings** request is less or equal to **tt:ColorSaturation/tt:Max** from the the **GetOptionsResponse** message AND
    - **tt:ColorSaturation** element value from the **SetImagingSettings** request is greater or equal to **tt:ColorSaturation/tt:Min** from the the **GetOptionsResponse** message AND
  - [S12] IF the **SetImagingSettings** request contains **tt:Contrast** element THEN:
    - The **GetOptionsResponse** message contains **tt:Contrast** element AND
    - **tt:Contrast** element value from the **SetImagingSettings** request is less or equal to **tt:Contrast/tt:Max** from the the **GetOptionsResponse** message AND
    - **tt:Contrast** element value from the **SetImagingSettings** request is greater or equal to

  - The **GetOptionsResponse** message contains **tt:Exposure** element AND
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:Mode** element value equal to **tt:Exposure/tt:Mode** element value from the **SetImagingSettings** request AND
- [S14] IF the **SetImagingSettings** request contains **tt:Exposure/tt:Priority** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:Priority** element AND
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:Priority** element value equal to **tt:Exposure/tt:Priority** element value from the **SetImagingSettings** request AND
- [S15] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MinExposureTime** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:MinExposureTime** element AND
  - **tt:Exposure/tt:MinExposureTime** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MinExposureTime/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MinExposureTime** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MinExposureTime/tt:Min** from the the **GetOptionsResponse** message AND
- [S16] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MaxExposureTime** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:MaxExposureTime** element AND
  - **tt:Exposure/tt:MaxExposureTime** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MaxExposureTime/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MaxExposureTime** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MaxExposureTime/tt:Min** from the the **GetOptionsResponse** message AND
- [S17] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MinGain** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:MinGain** element AND
  - **tt:Exposure/tt:MinGain** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MinGain/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MinGain** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MinGain/tt:Min** from the the **GetOptionsResponse** message AND
- [S18] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MaxGain** element THEN:

  - **tt:Exposure/tt:MaxGain** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MaxGain/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MaxGain** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MaxGain/tt:Min** from the the **GetOptionsResponse** message AND
- [S19] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MinIris** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:MinIris** element AND
  - **tt:Exposure/tt:MinIris** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MinIris/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MinIris** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MinIris/tt:Min** from the the **GetOptionsResponse** message AND
- [S20] IF the **SetImagingSettings** request contains **tt:Exposure/tt:MaxIris** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:MaxIris** element AND
  - **tt:Exposure/tt:MaxIris** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:MaxIris/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:MaxIris** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:MaxIris/tt:Min** from the the **GetOptionsResponse** message AND
- [S21] IF the **SetImagingSettings** request contains **tt:Exposure/tt:ExposureTime** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:ExposureTime** element AND
  - **tt:Exposure/tt:ExposureTime** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:ExposureTime/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:ExposureTime** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:ExposureTime/tt:Min** from the the **GetOptionsResponse** message AND
- [S22] IF the **SetImagingSettings** request contains **tt:Exposure/tt:Gain** element THEN:
  - The **GetOptionsResponse** message contains **tt:Exposure/tt:Gain** element AND
  - **tt:Exposure/tt:Gain** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:Gain/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:Gain** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:Gain/tt:Min** from the the **GetOptionsResponse** message AND
- [S23] IF the **SetImagingSettings** request contains **tt:Exposure/tt:Iris** element THEN:

  - **tt:Exposure/tt:Iris** element value from the **SetImagingSettings** request is less or equal to **tt:Exposure/tt:Iris/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Exposure/tt:Iris** element value from the **SetImagingSettings** request is greater or equal to **tt:Exposure/tt:Iris/tt:Min** from the the **GetOptionsResponse** message AND
- [S24] IF the **SetImagingSettings** request contains **tt:Focus** element THEN:
  - The **GetOptionsResponse** message contains **tt:Focus** element AND
  - The **GetOptionsResponse** message contains **tt:Focus/tt:AutoFocusModes** element value equal to **tt:Focus/tt:AutoFocusMode** element value from the **SetImagingSettings** request AND
- [S25] IF the **SetImagingSettings** request contains **tt:Focus/tt:DefaultSpeed** element THEN:
  - The **GetOptionsResponse** message contains **tt:Focus/tt:DefaultSpeed** element AND
  - **tt:Focus/tt:DefaultSpeed** element value from the **SetImagingSettings** request is less or equal to **tt:Focus/tt:DefaultSpeed/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Focus/tt:DefaultSpeed** element value from the **SetImagingSettings** request is greater or equal to **tt:Focus/tt:DefaultSpeed/tt:Min** from the the **GetOptionsResponse** message AND
- [S26] IF the **SetImagingSettings** request contains **tt:Focus/tt:NearLimit** element THEN:
  - The **GetOptionsResponse** message contains **tt:Focus/tt:NearLimit** element AND
  - **tt:Focus/tt:NearLimit** element value from the **SetImagingSettings** request is less or equal to **tt:Focus/tt:NearLimit/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Focus/tt:NearLimit** element value from the **SetImagingSettings** request is greater or equal to **tt:Focus/tt:NearLimit/tt:Min** from the the **GetOptionsResponse** message AND
- [S27] IF the **SetImagingSettings** request contains **tt:Focus/tt:FarLimit** element THEN:
  - The **GetOptionsResponse** message contains **tt:Focus/tt:FarLimit** element AND
  - **tt:Focus/tt:FarLimit** element value from the **SetImagingSettings** request is less or equal to **tt:Focus/tt:FarLimit/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Focus/tt:FarLimit** element value from the **SetImagingSettings** request is greater or equal to **tt:Focus/tt:FarLimit/tt:Min** from the the **GetOptionsResponse** message AND
- [S28] IF the **SetImagingSettings** request contains **tt:IrCutFilter** element THEN:
  - The **GetOptionsResponse** message contains **tt:IrCutFilterModes** element AND
  - The **GetOptionsResponse** message contains **tt:IrCutFilterModes** element value equal to **tt:IrCutFilter** element value from the **SetImagingSettings** request AND
- [S29] IF the **SetImagingSettings** request contains **tt:Sharpness** element THEN:
  - The **GetOptionsResponse** message contains **tt:Sharpness** element AND

- **tt:Sharpness** element value from the **SetImagingSettings** request is greater or equal to **tt:Sharpness/tt:Min** from the the **GetOptionsResponse** message AND
- [S30] IF the **SetImagingSettings** request contains **tt:WideDynamicRange** element THEN:
  - The **GetOptionsResponse** message contains **tt:WideDynamicRange** element AND
  - The **GetOptionsResponse** message contains **tt:WideDynamicRange/tt:Mode** element value equal to **tt:WideDynamicRange/tt:Mode** element value from the **SetImagingSettings** request AND
- [S31] IF the **SetImagingSettings** request contains **tt:WideDynamicRange/tt:Level** element THEN:
  - The **GetOptionsResponse** message contains **tt:WideDynamicRange/tt:Level** element AND
  - **tt:WideDynamicRange/tt:Level** element value from the **SetImagingSettings** request is less or equal to **tt:WideDynamicRange/tt:Level/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:WideDynamicRange/tt:Level** element value from the **SetImagingSettings** request is greater or equal to **tt:WideDynamicRange/tt:Level/tt:Min** from the the **GetOptionsResponse** message AND
- [S32] IF the **SetImagingSettings** request contains **tt:WhiteBalance** element THEN:
  - The **GetOptionsResponse** message contains **tt:WhiteBalance** element AND
  - The **GetOptionsResponse** message contains **tt:WhiteBalance/tt:Mode** element value equal to **tt:WhiteBalance/tt:Mode** element value from the **SetImagingSettings** request AND
- [S33] IF the **SetImagingSettings** request contains **tt:WhiteBalance/tt:CrGain** element THEN:
  - The **GetOptionsResponse** message contains **tt:WhiteBalance/tt:YrGain** element AND
  - **tt:WhiteBalance/tt:CrGain** element value from the **SetImagingSettings** request is less or equal to **tt:WhiteBalance/tt:YrGain/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:WhiteBalance/tt:CrGain** element value from the **SetImagingSettings** request is greater or equal to **tt:WhiteBalance/tt:YrGain/tt:Min** from the the **GetOptionsResponse** message AND
- [S34] IF the **SetImagingSettings** request contains **tt:WhiteBalance/tt:CbGain** element THEN:
  - The **GetOptionsResponse** message contains **tt:WhiteBalance/tt:YbGain** element AND
  - **tt:WhiteBalance/tt:CbGain** element value from the **SetImagingSettings** request is less or equal to **tt:WhiteBalance/tt:YbGain/tt:Max** from the the **GetOptionsResponse**

  - **tt:WhiteBalance/tt:CbGain** element value from the **SetImagingSettings** request is greater or equal to **tt:WhiteBalance/tt:YbGain/tt:Min** from the the **GetOptionsResponse** message AND
- [S35] IF the **SetImagingSettings** request contains **tt:Extension/tt:ImageStabilization** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:ImageStabilization** element AND
  - The **GetOptionsResponse** message contains **tt:Extension/tt:ImageStabilization/tt:Mode** element value equal to **tt:Extension/tt:ImageStabilization/tt:Mode** element value from the **SetImagingSettings** request AND
- [S36] IF the **SetImagingSettings** request contains **tt:Extension/tt:ImageStabilization/tt:Level** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:ImageStabilization/tt:Level** element AND
  - **tt:Extension/tt:ImageStabilization/tt:Level** element value from the **SetImagingSettings** request is less or equal to **tt:Extension/tt:ImageStabilization/tt:Level/tt:Max** from the the **GetOptionsResponse** message AND
  - **tt:Extension/tt:ImageStabilization/tt:Level** element value from the **SetImagingSettings** request is greater or equal to **tt:Extension/tt:ImageStabilization/tt:Level/tt:Min** from the the **GetOptionsResponse** message AND
- [S37] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment** element AND
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryType** element value equal to each **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryType** element value from the **SetImagingSettings** request AND
- [S38] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryOffset** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryOffset** element with value equal to true AND
  - **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryOffset** element value from the **SetImagingSettings** request is less or equal to +1.0 AND
  - **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:BoundaryOffset** element value from the **SetImagingSettings** request is greater or equal to -1.0 AND
- [S39] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTime** element THEN:

  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTimeRange** element AND
  - **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTime** element value from the **SetImagingSettings** request is less or equal to **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTimeRange** from the the **GetOptionsResponse** message AND
  - **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTime** element value from the **SetImagingSettings** request is greater or equal to **tt:Extension/tt:Extension/tt:IrCutFilterAutoAdjustment/tt:ResponseTimeRange** from the the **GetOptionsResponse** message AND
- [S40] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensation** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensationOptions** element AND
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensationOptions/tt:Mode** element value equal to **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensation/tt:Mode** element value from the **SetImagingSettings** request AND
- [S41] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensation/tt:Level** element THEN:
  - The **GetOptionsResponse** message contains **>tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensationOptions/tt:Level** element with value equal to true AND
  - **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensation/tt:Level** element value from the **SetImagingSettings** request is less or equal to +1.0 AND
  - **tt:Extension/tt:Extension/tt:Extension/tt:ToneCompensation/tt:Level** element value from the **SetImagingSettings** request is greater or equal to 0.0 AND
- [S42] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:Defogging** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:Extension/tt:DefoggingOptions** element AND
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:Extension/tt:DefoggingOptions/tt:Mode** element value equal to **tt:Extension/tt:Extension/tt:Extension/tt:Defogging/tt:Mode** element value from the **SetImagingSettings** request AND
- [S43] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:Defogging/tt:Level** element THEN:
  - The **GetOptionsResponse** message contains **>tt:Extension/tt:Extension/tt:Extension/tt:DefoggingOptions/tt:Level** element with value equal to true AND

- - **tt:Extension/tt:Extension/tt:Extension/tt:Defogging/tt:Level** element value from the **SetImagingSettings** request is less or equal to +1.0 AND
  - **tt:Extension/tt:Extension/tt:Extension/tt:Defogging/tt:Level** element value from the **SetImagingSettings** request is greater or equal to 0.0 AND
- [S44] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:NoiseReduction** element THEN:
  - The **GetOptionsResponse** message contains **tt:Extension/tt:Extension/tt:Extension/tt:NoiseReductionOptions** element AND
- [S45] IF the **SetImagingSettings** request contains **tt:Extension/tt:Extension/tt:Extension/tt:NoiseReduction/tt:Level** element THEN:
  - The **GetOptionsResponse** message contains **>tt:Extension/tt:Extension/tt:Extension/tt:NoiseReductionOptions/tt:Level** element with value equal to true AND
  - **tt:Extension/tt:Extension/tt:Extension/tt:NoiseReduction/tt:Level** element value from the **SetImagingSettings** request is less or equal to +1.0 AND
  - **tt:Extension/tt:Extension/tt:Extension/tt:NoiseReduction/tt:Level** element value from the **SetImagingSettings** request is greater or equal to 0.0.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_imaging_settings.get_imaging_settings

8 Focus Move Capabilities Test Cases

Validated Feature: get_move_options

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

8.1 Expected Scenarios Under Test:

1. Client connects to Device to get focus move capabilities.
2. Client is considered as supporting Focus Move Capabilities if the following conditions are met:
   - Client is able to retrieve a focus move options using **GetMoveOptions** operation AND
3. Client is considered as NOT supporting Focus Move Capabilities if ANY of the following is TRUE:
   - No valid responses for **GetMoveOptions** request OR

8.2 GET FOCUS MOVE OPTIONS

Test Label: Get Move Options

Test Case ID: GETMOVEOPTIONS-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Move Options

Test Purpose: To verify that Client is able retrive focus move capabilities from Device using the **GetMoveOptions** operation.

Pre-Requisite:

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

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

1. Client invokes **GetMoveOptions** request message to retrieve focus move options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetMoveOptionsResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_move_options.get_move_options

9 Focus Control Test Cases

Validated Feature: focus_control

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

9.1 Expected Scenarios Under Test:

1. Client connects to Device to control focus.
2. Client is considered as supporting Focus Control if the following conditions are met:
    - Client supports get_move_options feature AND
    - Client is able to invoke Absolute OR Relative OR Continuous focus move using **Move** operation AND
    - If Client is able to invoke Continuous focus move Client is able to invoke stop focus move using **Stop** operation.
3. Client is considered as NOT supporting Focus Control if ANY of the following is TRUE:
    - Client does not support get_move_options feature OR
    - No valid responses for **Move** request OR
    - **Move** request contains settings which does not correspong to **GetMoveOptions** message for the same video source token OR
    - No valid responses for **Stop** request if **Stop** request is supported by the Client OR
    - **Stop** request is not supported, in the case Continuous focus move is supported by the Client.

9.2 ABSOLUTE FOCUS MOVE

Test Label: Focus Control - Absolute Focus Move

Test Case ID: FOCUSCONTROL-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Absolute Focus Move

Test Purpose: To verify that Client is able retrive absolute focus move on Device using the **Move** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **Move** operation with **tt:Absolute** element present.
- Device supports Imaging Service.

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

1. Client invokes **GetMoveOptions** request message to retrieve focus move options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetMoveOptionsResponse** message.
3. If **GetMoveOptionsResponse** message contains **tt:Absolute** element Client invokes **Move** request message for specified video source with **tt:Absolute** element with parameters which are correspond to the resieved focus move options to start absolute focus movement on the Device.
4. Device responses with code HTTP 200 OK and **MoveResponse** message.

Test Result:

PASS -

- Client **Move** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Move** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **timg:Move** AND
  - [S2] It contains **timg:Focus/tt:Absolute** element AND

- Device response on the **Move** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **timg:MoveResponse** AND

- There is a Client **GetMoveOptions** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **Move** request AND
  - [S6] It invoked before the Client **Move** request AND
  - [S7] **timg:VideoSourceToken** element value is equal to **timg:VideoSourceToken** element from the **Move** request AND

- Device response on the **GetMoveOptions** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **timg:GetMoveOptionsResponse** AND
  - [S10] It contains **timg:MoveOptions\tt:Absolute** element AND

- Settings from the **Move** request corresponds options recieved in the **GetMoveOptionsResponse** message:
  - [S11] **timg:Focus/tt:Absolute/tt:Position** element value from the **Move** request is less or equal to **timg:MoveOptions/tt:Absolute/tt:Position/tt:Max** from the the **GetMoveOptionsResponse** message AND
  - [S12] **timg:Focus/tt:Absolute/tt:Position** element value from the **Move** request is greater or equal to **timg:MoveOptions/tt:Absolute/tt:Position/tt:Min** from the the **GetMoveOptionsResponse** message AND
  - [S13] IF the **Move** request contains **timg:Focus/tt:Absolute/tt:Speed** element THEN:
    - The **GetMoveOptionsResponse** message contains **timg:MoveOptions/tt:Absolute/tt:Speed** element AND
    - **timg:Focus/tt:Absolute/tt:Speed** element value from the **Move** request is less or equal to **timg:MoveOptions/tt:Absolute/tt:Speed/tt:Max** from the the **GetMoveOptionsResponse** message AND
    - **timg:Focus/tt:Absolute/tt:Speed** element value from the **Move** request is greater or equal to **timg:MoveOptions/tt:Absolute/tt:Speed/tt:Min** from the the **GetMoveOptionsResponse** message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: focus_control.absolute_focus_move

9.3 RELATIVE FOCUS MOVE

Test Label: Focus Control - Relative Focus Move

Test Case ID: FOCUSCONTROL-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Relative Focus Move

Test Purpose: To verify that Client is able retrive relative focus move on Device using the **Move** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **Move** operation with **tt:Relative** element present.
- Device supports Imaging Service.

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

1. Client invokes **GetMoveOptions** request message to retrieve focus move options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetMoveOptionsResponse** message.
3. If **GetMoveOptionsResponse** message contains **tt:Relative** element Client invokes **Move** request message for specified video source with **tt:Relative** element with parameters which are correspond to the resieved focus move options to start relative focus movement on the Device.
4. Device responses with code HTTP 200 OK and **MoveResponse** message.

Test Result:

PASS -

- Client **Move** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Move** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **timg:Move** AND
  - [S2] It contains **timg:Focus/tt:Relative** element AND

- Device response on the **Move** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **timg:MoveResponse** AND
- There is a Client **GetMoveOptions** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **Move** request AND
  - [S6] It invoked before the Client **Move** request AND
  - [S7] **timg:VideoSourceToken** element value is equal to **timg:VideoSourceToken** element from the **Move** request AND
- Device response on the **GetMoveOptions** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **timg:GetMoveOptionsResponse** AND
  - [S10] It contains **timg:MoveOptions\tt:Relative** element AND
- Settings from the **Move** request corresponds options recieved in the **GetMoveOptionsResponse** message:
  - [S11] **timg:Focus/tt:Relative/tt:Distance** element value from the **Move** request is less or equal to **timg:MoveOptions/tt:Relative/tt:Distance/tt:Max** from the the **GetMoveOptionsResponse** message AND
  - [S12] **timg:Focus/tt:Relative/tt:Distance** element value from the **Move** request is greater or equal to **timg:MoveOptions/tt:Relative/tt:Distance/tt:Min** from the the **GetMoveOptionsResponse** message AND
  - [S13] IF the **Move** request contains **timg:Focus/tt:Relative/tt:Speed** element THEN:
    - The **GetMoveOptionsResponse** message contains **timg:MoveOptions/tt:Relative/tt:Speed** element AND
    - **timg:Focus/tt:Relative/tt:Speed** element value from the **Move** request is less or equal to **timg:MoveOptions/tt:Relative/tt:Speed/tt:Max** from the the **GetMoveOptionsResponse** message AND
    - **timg:Focus/tt:Relative/tt:Speed** element value from the **Move** request is greater or equal to **timg:MoveOptions/tt:Relative/tt:Speed/tt:Min** from the the **GetMoveOptionsResponse** message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: focus_control.relative_focus_move

9.4 CONTINUOUS FOCUS MOVE

Test Label: Focus Control - Continuous Focus Move

Test Case ID: FOCUSCONTROL-3

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Continuous Focus Move

Test Purpose: To verify that Client is able retrive continuous focus move on Device using the **Move** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **Move** operation with **tt:Continuous** element present.
- Device supports Imaging Service.

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

1. Client invokes **GetMoveOptions** request message to retrieve focus move options for specified video source from the Device.
2. Device responses with code HTTP 200 OK and **GetMoveOptionsResponse** message.
3. If **GetMoveOptionsResponse** message contains **tt:Continuous** element Client invokes **Move** request message for specified video source with **tt:Relative** element with parameters which are correspond to the resieved focus move options to start continuous focus movement on the Device.
4. Device responses with code HTTP 200 OK and **MoveResponse** message.

Test Result:

PASS -

- Client **Move** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Move** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **timg:Move** AND
  - [S2] It contains **timg:Focus/tt:Continuous** element AND

- Device response on the **Move** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **timg:MoveResponse** AND
- There is a Client **GetMoveOptions** request in Test Procedure fulfills the following requirements:
  - [S5] It invoked for the same Device as for the Client **Move** request AND
  - [S6] It invoked before the Client **Move** request AND
  - [S7] **timg:VideoSourceToken** element value is equal to **timg:VideoSourceToken** element from the **Move** request AND
- Device response on the **GetMoveOptions** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **timg:GetMoveOptionsResponse** AND
  - [S10] It contains **timg:MoveOptions\tt:Continuous** element AND
- Settings from the **Move** request corresponds options recieved in the **GetMoveOptionsResponse** message:
  - [S11] **timg:Focus/tt:Continuous/tt:Speed** element value from the **Move** request is less or equal to **timg:MoveOptions/tt:Continuous/tt:Speed/tt:Max** from the the **GetMoveOptionsResponse** message AND
  - [S12] **timg:Focus/tt:Continuous/tt:Speed** element value from the **Move** request is greater or equal to **timg:MoveOptions/tt:Continuous/tt:Speed/tt:Min** from the the **GetMoveOptionsResponse** message.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: focus_control.continuous_focus_move

9.5 STOP

Test Label: Focus Control - Stop

Test Case ID: FOCUSCONTROL-4

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Stop

Test Purpose: To verify that Client is able retrivefocus move options fromon Device using the **Stop** operation.

Pre-Requisite:

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

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

1. Client invokes **Stop** request message to stop focus move for specified video source for the Device.
2. Device responses with code HTTP 200 OK and **StopResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: focus_control.stop