

ONVIF™
ONVIF Imaging Test Specification
Version 17.01
January 2017



© 2017 by ONVIF, Inc. All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

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

Ver.	Date	Description
11.12	22nd/Dec, 2011	First issue
12.06	18 th /June,2012	Imaging Service Capabilities test cases have been added.
13.06	June,2013	The following test cases was added: IMAGING COMMAND SETIMAGINGSETTINGS
13.12	Dec, 2013	Minor changes
14.06	June, 2014	IMAGING COMMAND SETIMAGINGSETTINGS test case was changed
14.12	Dec, 2014	IMAGING COMMAND SETIMAGINGSETTINGS test case was changed
16.06	May 2016	4.1.8 IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL FEATURES has been added.
16.07	July 5 2016	Removed Exposure.Window testing, several comments from Sony have been implemented
16.07	July 13, 2016	4.1.8 - Test sequence updated
16.07	July 26, 2016	Comments implemented
16.07	August 8, 2016	Comments from Hiroyuki Sano
16.09	September, 2016	IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL FEATURES test case was changed Annex A.1 was changed.
17.01	December, 2016	Test Sequence updated for IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL FEATURES test case.



Table of Contents

1	Introduction	6
1.1	Scope	6
1.1.1	Imaging Setting.....	6
1.1.2	Focus Move	7
2	Terms and Definitions	8
2.1	Definitions.....	8
3	Test Overview.....	9
3.1	Test Setup	9
3.1.1	Network Configuration for device under test	9
3.2	Prerequisites	10
3.3	Test Policy	10
3.3.1	Imaging Setting.....	10
3.3.2	Focus Move	10
4	Imaging Test Cases.....	11
4.1	Imaging Settings.....	11
4.1.1	IMAGING COMMAND GETIMAGINGSETTINGS	11
4.1.2	IMAGING COMMAND GETOPTIONS	12
4.1.3	IMAGING COMMAND SETIMAGINGSETTINGS – INVALID SETTINGS	14
4.1.4	IMAGING COMMAND GETIMAGINGSETTINGS – INVALID VIDEOSOURCETOKEN	16
4.1.5	IMAGING COMMAND GETOPTIONS – INVALID VIDEOSOURCETOKEN	17
4.1.6	IMAGING COMMAND SETIMAGINGSETTINGS – INVALID VIDEOSOURCETOKEN	19
4.1.7	IMAGING COMMAND SETIMAGINGSETTINGS	20
4.1.8	IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL FEATURES	54
4.2	Focus Move	65
4.2.1	IMAGING COMMAND GETMOVEOPTIONS.....	65
4.2.2	IMAGING COMMAND ABSOLUTE MOVE	66
4.2.3	IMAGING COMMAND ABSOLUTE MOVE – INVALID SETTINGS	68
4.2.4	IMAGING COMMAND RELATIVE MOVE	70
4.2.5	IMAGING COMMAND RELATIVE MOVE – INVALID SETTINGS	72
4.2.6	IMAGING COMMAND CONTINUOUS MOVE	74
4.2.7	IMAGING COMMAND CONTINUOUS MOVE – INVALID SETTINGS	76
4.2.8	IMAGING COMMAND MOVE – UNSUPPORTED MOVE	78
4.2.9	IMAGING COMMAND GETSTATUS.....	79
4.2.10	IMAGING COMMAND STOP	81
4.2.11	IMAGING COMMAND GETMOVEOPTIONS – INVALID VIDEOSOURCETOKEN.....	82
4.2.12	IMAGING COMMAND MOVE – INVALID VIDEOSOURCETOKEN	83
4.2.13	IMAGING COMMAND GETSTATUS – INVALID VIDEOSOURCETOKEN	84
4.2.14	IMAGING COMMAND STOP – INVALID VIDEOSOURCETOKEN	86



4.3	Capabilities	88
4.3.1	IMAGING SERVICE CAPABILITIES	88
4.3.2	GET SERVICES AND GET IMAGING SERVICE CAPABILITIES CONSISTANCY	89
Annex A	91
A.1	Get Video Sources.....	91
A.2	Name and Token Parameters	91



1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases necessary to verify the [ONVIF Network Interface Specs] and [ONVIF Conformance] requirements⁴ and the test cases are to be basic inputs for some Profile specification requirements. It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.

This ONVIF Imaging Test Specification acts as a supplementary document to the [ONVIF Network Interface Specs], illustrating test cases required to be executed and passed. Also, this specification acts as an input document to the development of test tool which will be used to test the ONVIF device implementation conformance towards ONVIF standard. This test tool is referred as ONVIF Client hereafter.

1.1 Scope

This ONVIF Imaging Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification is to provide the test cases to test individual requirements of ONVIF devices according to ONVIF Imaging service which is defined in [ONVIF Network Interface Specs].

The principal intended purposes are:

1. Provide self-assessment tool for implementations.
2. Provide comprehensive test suite coverage for [ONVIF Network Interface Specs].

This specification does not address the following.

1. Product use cases and non-functional (performance and regression) testing.
2. SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
3. Network protocol implementation Conformance test for HTTPS, HTTP, RTP and RTSP protocols.
4. Wi-Fi Conformance test

The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Network Interface Specs]; instead it would cover its subset.

This ONVIF Imaging Test Specification covers ONVIF Imaging service which is a functional block of [ONVIF Network Interface Specs]. The following sections give the brief overview and scope of each functional block.

1.1.1 Imaging Setting

Imaging Setting test cases are covered for verification to get and set imaging parameters for BacklightCompensation, Brightness, ColorSaturation, Sharpness, Contrast, Exposure, Focus, Ir cut filter, WhiteBalance, WideDynamicRange, Stabilization, IR cur filter auto adjustment, ToneCompensation, Defogging, and NoiseReduction. It means that **GetImagingSettings**, **GetOptions** and **SetImagingSettings** commands are covered by this test case.



1.1.2 Focus Move

Focus Move test cases are covered for verification of capabilities to move the focus lens. ***GetMoveOptions, Move, Stop and GetStatus*** commands are covered by this test case.



2 Terms and Definitions

2.1 Definitions

This section defines terms that are specific to the ONVIF Imaging Service and tests. For the list of applicable general terms and definitions, please see [ONVIF Base Test].

BacklightCompensation	Backlight Compensation
Brightness	Image brightness
ColorSaturation	Color saturation of the image
Contrast	Contrast of the image
Defogging	Clarify image contrast effect
Exposure	Exposure
Focus	Focus
Imaging Service	Services for exposure time, gain and white balance parameters among others.
Ir cut filter	Infrared cutoff filter
IR cut filter auto adjust	Alternate IR cut filter depend on the brightness automatically.
Sharpness	Sharpness of the video image
Stabilization	Stabilize images against device vibration
Tone compensation	Image contrast compensation appropriately
WhiteBalance	White balance
WideDynamicRange	Wide dynamic range

3 Test Overview

This section describes the test setup procedure and prerequisites needed, and the test policies that should be followed for test case execution.

3.1 Test Setup

3.1.1 Network Configuration for device under test

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 1)

Based on the individual test case requirements, some of the entities in the below setup may not be needed for the execution of those corresponding test cases.

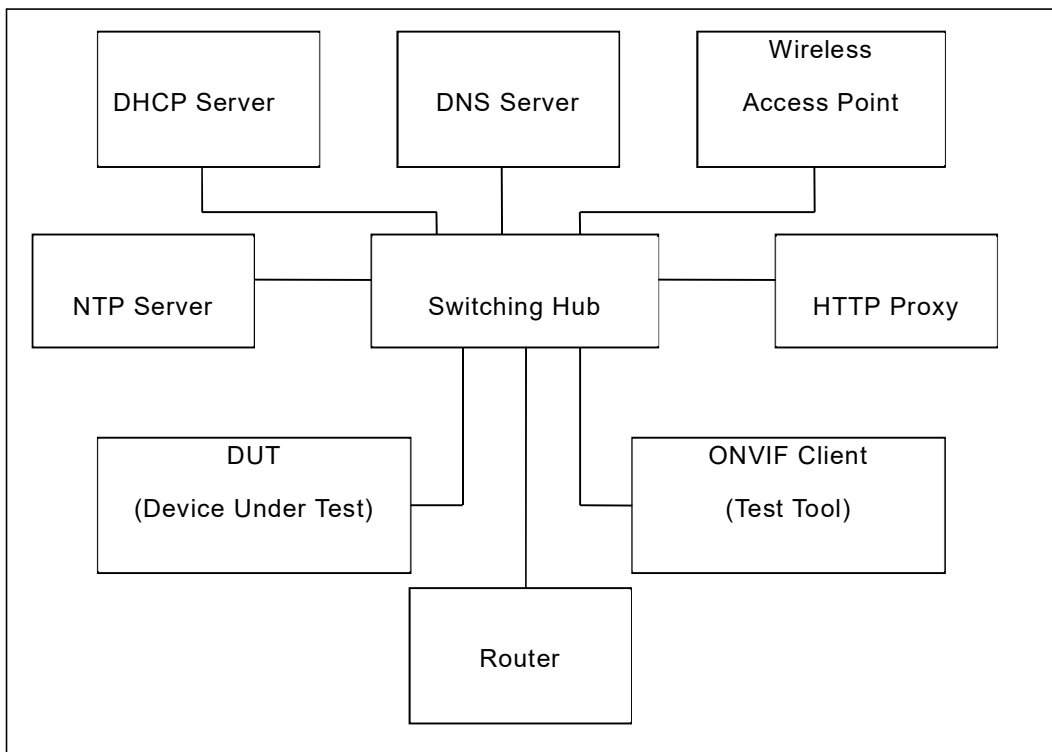


Figure 1: Test Configuration for DUT

DUT: ONVIF device to be tested. Hereafter, this is referred to as DUT (Device Under Test).

ONVIF Client (Test Tool): Tests are executed by this system and it controls the behaviour of the DUT. It handles both expected and unexpected behaviour.

HTTP Proxy: provides facilitation in case of RTP and RTSP tunnelling over HTTP.

Wireless Access Point: provides wireless connectivity to the devices that support wireless connection.

DNS Server: provides DNS related information to the connected devices.

DHCP Server: provides IPv4 Address to the connected devices.



NTP Server: provides time synchronization between ONVIF Client and DUT.

Switching Hub: provides network connectivity among all the test equipment in the test environment. All devices should be connected to the Switching Hub.

Router: provides router advertisements for IPv6 configuration.

3.2 Prerequisites

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

- The DUT shall be configured with an IPv4 address.
- The DUT shall be IP reachable [in the test configuration].
- The DUT shall be able to be discovered by the ONVIF Client Test Tool.
- The DUT shall be configured with the time i.e. manual configuration of UTC time and if NTP is supported by DUT then NTP time shall be synchronized with NTP Server.
- The DUT time and ONVIF Client Test tool time shall be synchronized with each other either manually or by common NTP server.

3.3 Test Policy

This section describes the test policies specific to the test case execution of each functional block.

The DUT shall adhere to the test policies defined in this section.

3.3.1 Imaging Setting

DUT needs to support either Media service or Device IO service because of obtaining Video Source.

DUT shall give the Imaging Service entry point by GetCapabilities command. And Media service or Device IO service entry point also shall be provided by GetCapabilities command.

DUT shall support at least one Video Source and return the configuration through GetVideoSources command of Media service or Device IO.

3.3.2 Focus Move

Requirement is same with 3.3.1 Imaging Setting. If DUT supports remote focus control, DUT shall indicate it by returning GetMoveOptionsResponse with at least one parameter.



4 Imaging Test Cases

4.1 Imaging Settings

4.1.1 IMAGING COMMAND GETIMAGINGSETTINGS

Test Label: GetImagingSettings Command Validation

Test Case ID: IMAGING-1-1-1

ONVIF Core Specification Coverage: Get imaging settings

Command Under Test: GetImagingSettings

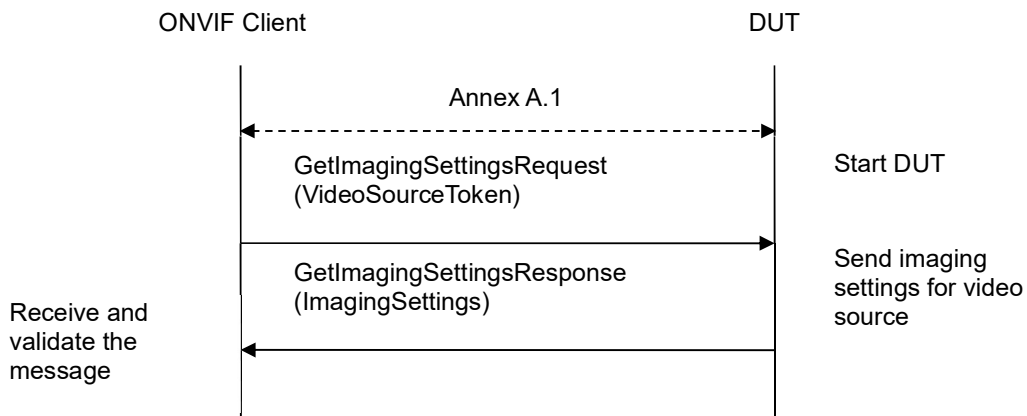
WSDL Reference: imaging.wsdl

Test Purpose: To verify the behaviour of GetImagingSettings command.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetImagingSettingsRequest message (VideoSourceToken) to retrieve current imaging settings for Video Source.
4. Verify the GetImagingSettingsResponse message from the DUT.
5. ONVIF Client will repeat steps 3-4 for other video sources supported by the DUT.

Test Result:

PASS –

DUT passed all assertions.



FAIL –

- The DUT did not send GetImagingSettingsResponse message.
- The DUT sent an invalid GetImagingSettingsResponse message.

4.1.2 IMAGING COMMAND GETOPTIONS

Test Label: Imaging DUT GetOptions Command Validation.

Test Case ID: IMAGING-1-1-3

ONVIF Core Specification Coverage: Get options

Command under Test: GetOptions

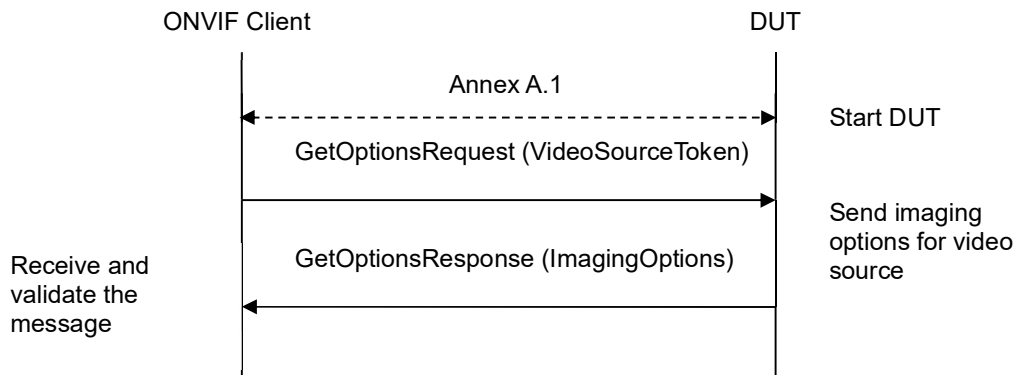
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetOptions command.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetOptionsRequest message (VideoSourceToken) to retrieve current imaging options for Video Source.
4. Verify the GetOptionsResponse message from the DUT.
5. ONVIF Client will repeat steps 3-4 for other video sources supported by the DUT.

Test Result:

PASS –



DUT passed all assertions.

FAIL –

The DUT did not send GetOptionsResponse message.

The DUT did not send a valid GetOptionsResponse message.

The DUT returned GetOptionsResponse message with Min value greater than Max for at least one setting defined by Min and Max boundary:

- BacklightCompensation.Level
- Brightness
- ColorSaturation
- Contrast
- Exposure.MinExposureTime
- Exposure.MaxExposureTime
- Exposure.MinGain
- Exposure.MaxGain
- Exposure.MinIris
- Exposure.MaxIris
- Exposure.ExposureTime
- Exposure.Gain
- Exposure.Iris
- Focus.DefaultSpeed
- Focus.NearLimit
- Focus.FarLimit
- Sharpness
- WideDynamicRange.Level
- WhiteBalance.YrGain
- WhiteBalance.YbGain

The DUT returned GetOptionsResponse message with the list of Exposure options, which does not correspond the list of possible Exposure.Mode:

- If only Exposure.Mode=Manual supported, then there shall be no:
 - Exposure.Priority.Window
 - Exposure.MinExposureTime



- Exposure.MaxExposureTime
 - Exposure.MinGain
 - Exposure.MaxGain
 - Exposure.MinIris
 - Exposure.MaxIris
- If only Exposure.Mode=Auto supported, then there shall be no Exposure.ExposureTime, Exposure.Gain, Exposure.Iris.

The DUT returned GetOptionsResponse with BacklightCompensation.Level option if only BacklightCompensation.Mode=OFF.

The DUT returned GetOptionsResponse with the list of Focus options that does not correspond the list of possible Focus.Mode:

- If only Focus.Mode=Manual supported, then there shall be no Focus.NearLimit, Focus.FarLimit.
- If only Focus.Mode=Auto supported, then there shall be no Focus.DefaultSpeed.

The DUT returned GetOptionsResponse with the list of WhiteBalance options that does not correspond the list of possible WhiteBalance.Mode:

- If only WhiteBalance.Mode=Auto supported, then there shall be no WhiteBalance.YrGain, WhiteBalance.YbGain.

The DUT returned GetOptionsResponse with the list of WideDynamicRange options that does not correspond the list of possible WideDynamicRange.Mode:

- If only WideDynamicRange.Mode=OFF supported, then there shall be no WideDynamicRange.Level.

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.1.3 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID SETTINGS

Test Label: Imaging DUT SetImagingSettings Command Validation (Invalid ImagingSettings).

Test Case ID: IMAGING-1-1-8

ONVIF Core Specification Coverage: Set imaging settings

Command under Test: SetImagingSettings

WSDL Reference: imaging.wsdl

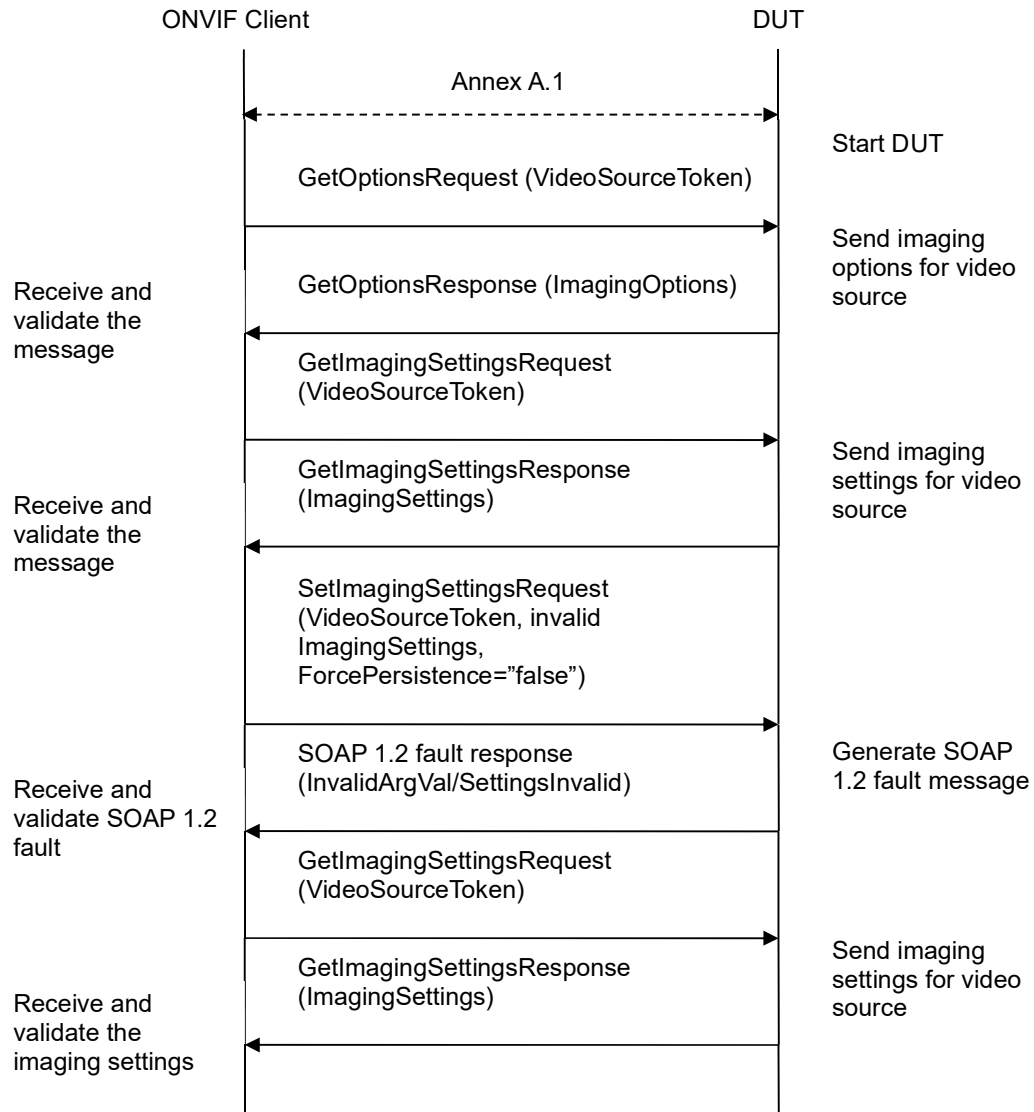
Test Propose: To verify the behavior of SetImagingSettings command in case of invalid ImagingSettings.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT



Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetOptionsRequest message (VideoSourceToken) to retrieve current imaging options for Video Source.
4. Verify the GetOptionsResponse message from the DUT.
5. ONVIF Client will invoke GetImagingSettingsRequest message (VideoSourceToken) to retrieve current imaging settings.



6. Verify the GetImagingSettingsResponse message from the DUT.
7. ONVIF Client will invoke SetImagingSettingsRequest message (VideoSourceToken, all supported parameters from GetOptionsResponse set to invalid value, force persistence = "false").
8. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/SettingsInvalid).
9. ONVIF Client will invoke GetImagingSettingsRequest message (VideoSourceToken) to retrieve current imaging settings.
10. Verify the GetImagingSettingsResponse message from the DUT.
11. Verify that ImagingSettings is the same as GetImagingSettingsResponse message at step 6.
12. ONVIF Client will repeat steps 3-11 for other video sources supported by the DUT.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send GetOptionsResponse message.

The DUT did not send a valid GetOptionsResponse message.

The DUT did not send GetImagingSettingsResponse message.

The DUT did not send a valid GetImagingSettingsResponse message.

The DUT did not send SOAP 1.2 fault message for SetImagingSettingsRequest message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

The DUT modified imaging settings by invoked SetImagingSettingsRequest message.

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.1.4 IMAGING COMMAND GETIMAGINGSETTINGS – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT GetImagingSettings Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-1-1-10

ONVIF Core Specification Coverage: Get imaging settings

Command Under Test: GetImagingSettings

WSDL Reference: imaging.wsdl

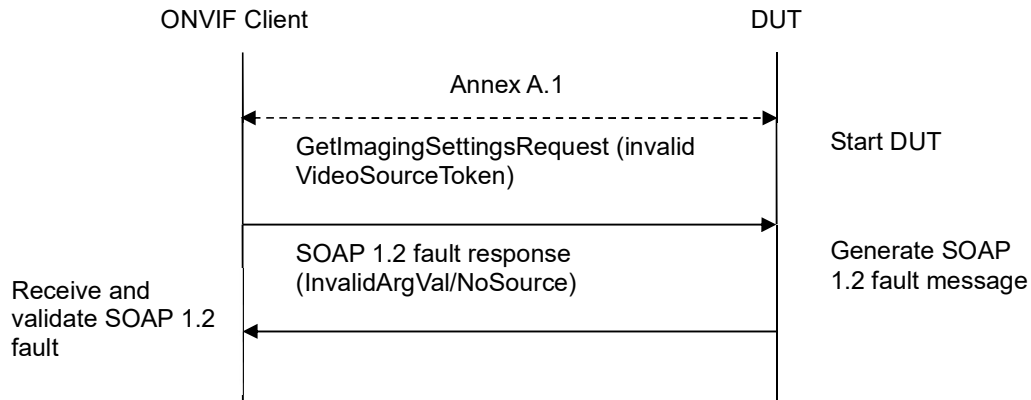
Test Purpose: To verify the behaviour of GetImagingSettings command in case of invalid VideoSourceToken.



Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetImagingSettingsRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.

4.1.5 IMAGING COMMAND GETOPTIONS – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT GetOptions Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-1-1-11

ONVIF Core Specification Coverage: Get options



Command under Test: GetOptions

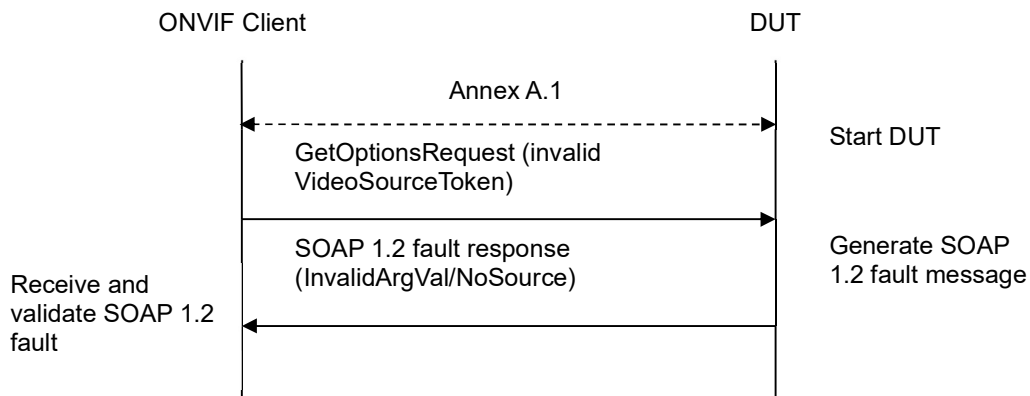
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetOptions command in case of invalid VideoSourceToken.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetOptionsRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.



4.1.6 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT SetImagingSettings Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-1-1-12

ONVIF Core Specification Coverage: Set imaging settings

Command under Test: SetImagingSettings

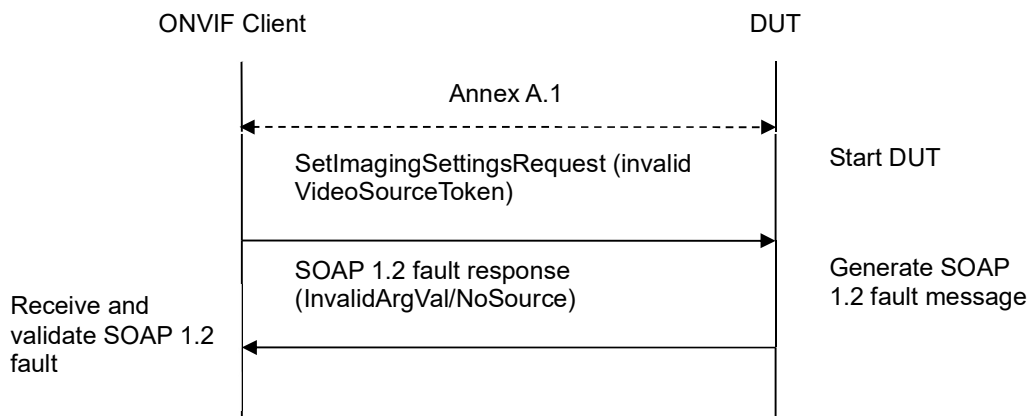
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of SetImagingSettings command in case of invalid VideoSourceToken.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke SetImagingSettingsRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource).

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).



Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.

4.1.7 IMAGING COMMAND SETIMAGINGSETTINGS

Test Label: Imaging DUT SetImagingSettings Command Validation.

Test Case ID: DRAFT-IMAGING-1-1-14

ONVIF Core Specification Coverage: Set imaging settings

Command under Test: SetImagingSettings

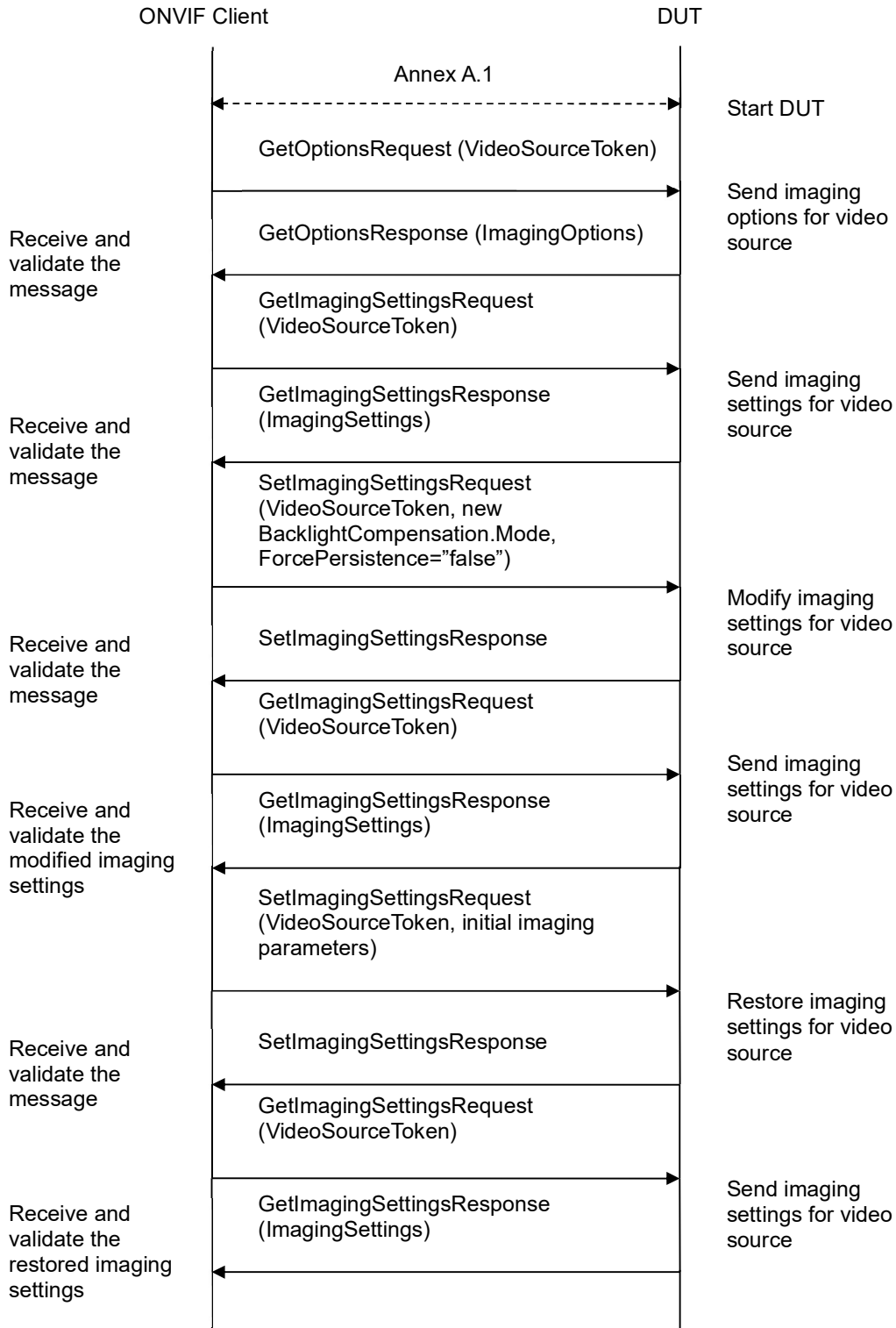
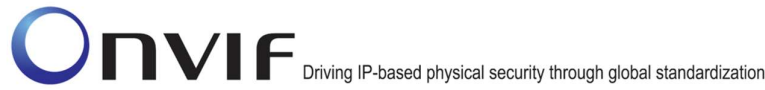
WSDL Reference: imaging.wsdl

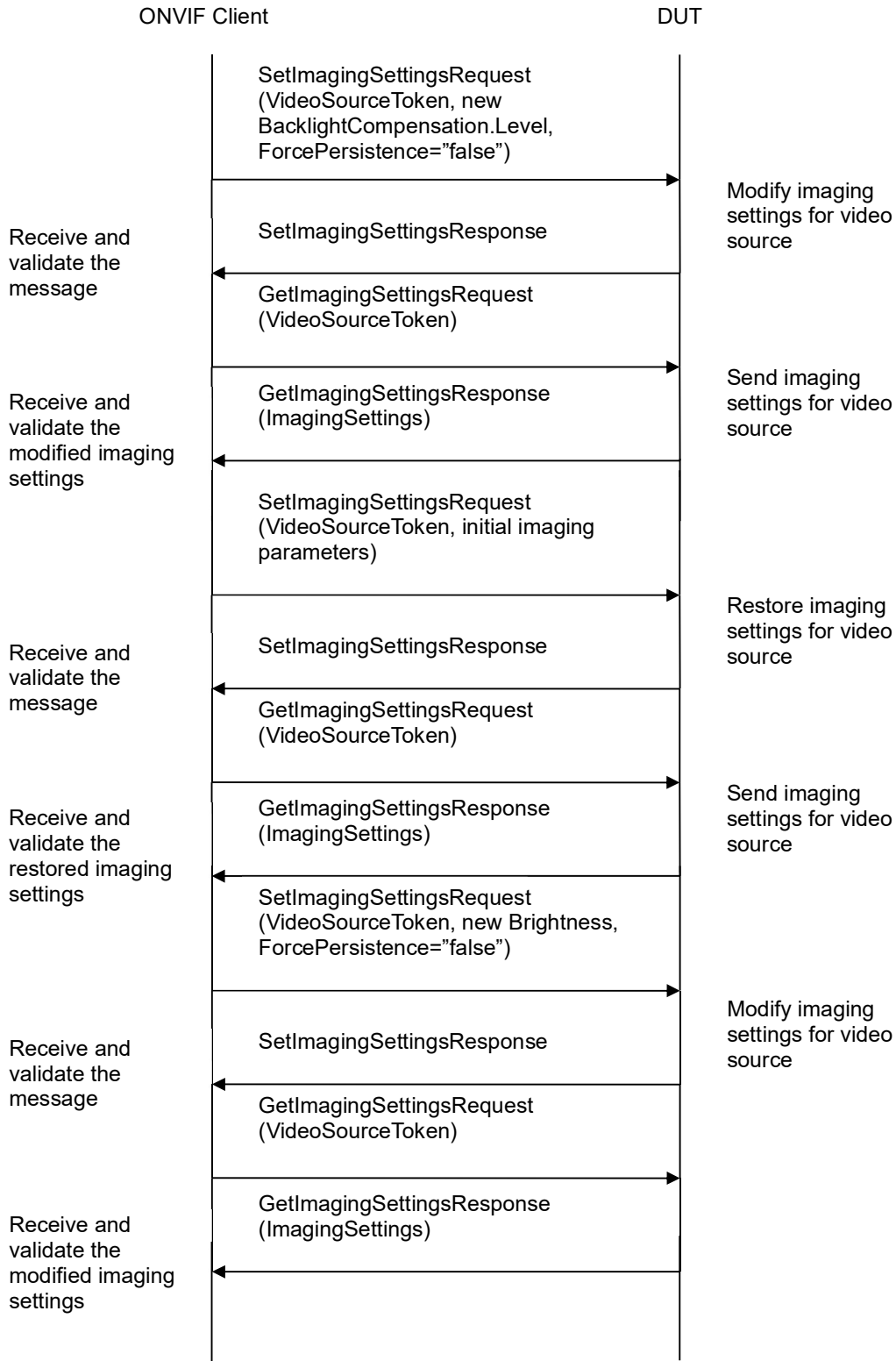
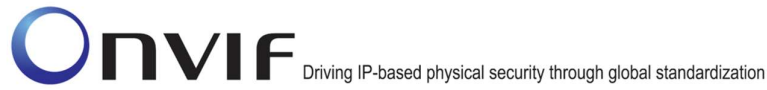
Test Propose: To verify the behaviour of SetImagingSettings command and to verify DUT Imaging Setting.

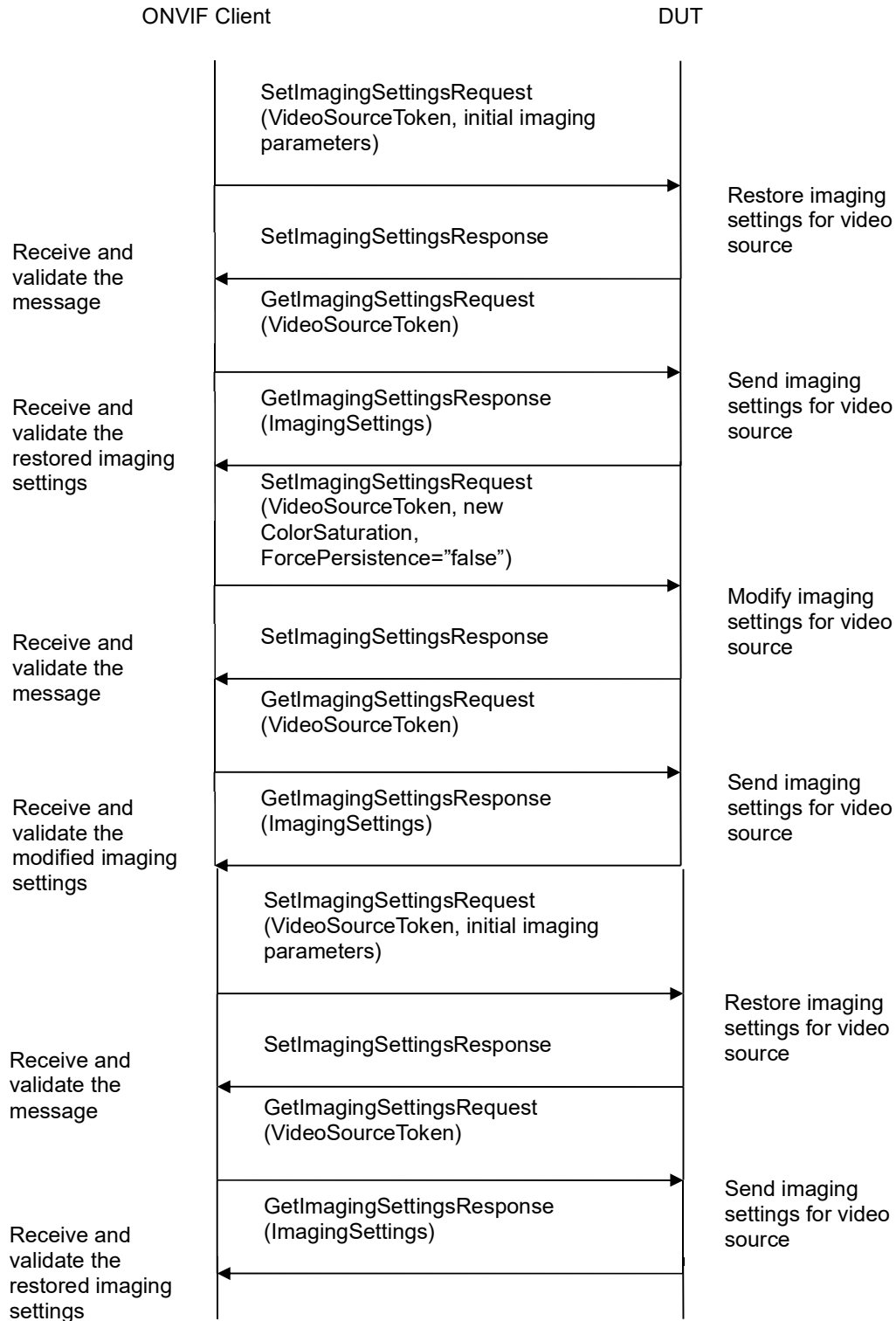
Pre-Requisite: None

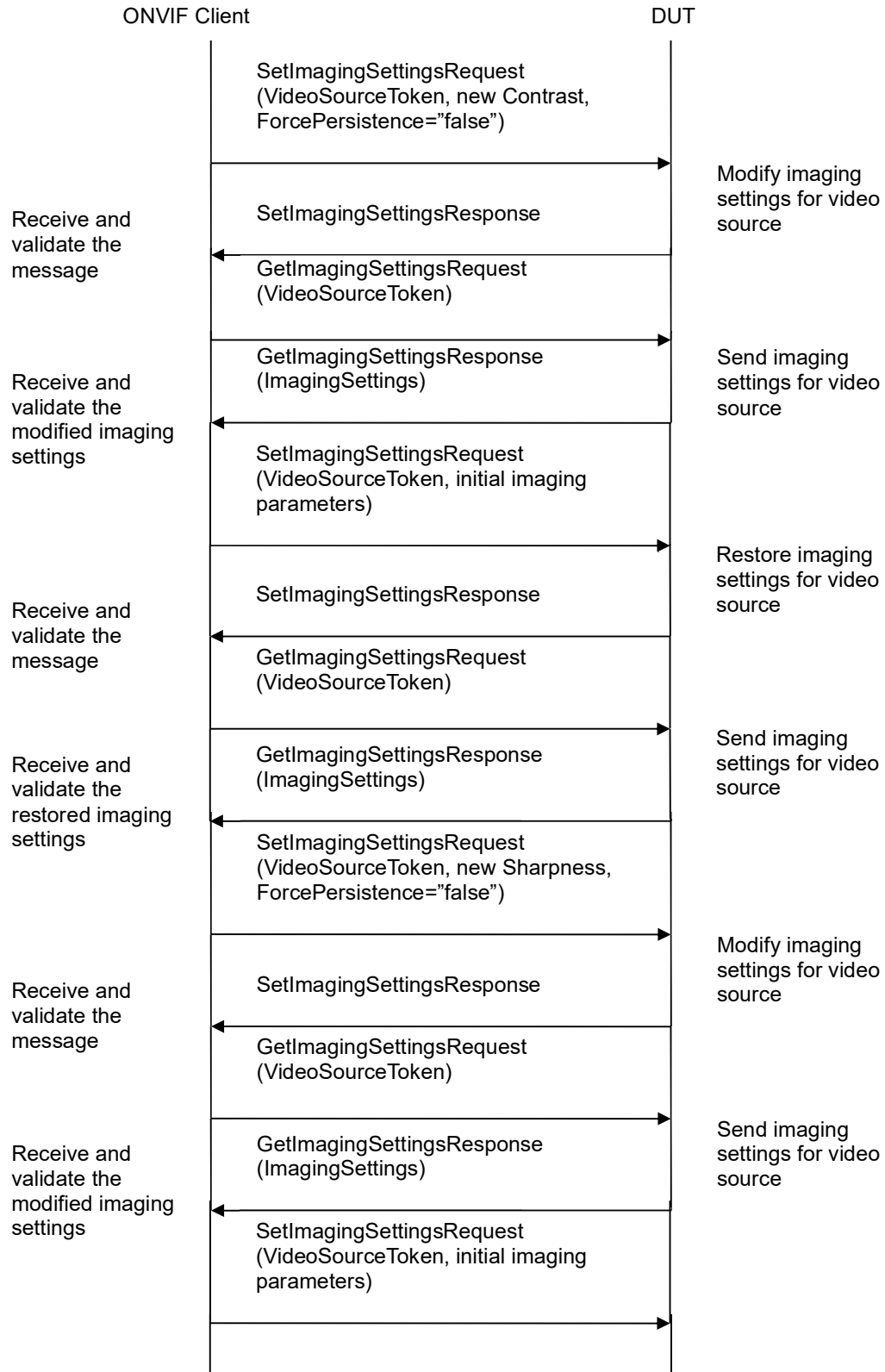
Test Configuration: ONVIF Client and DUT

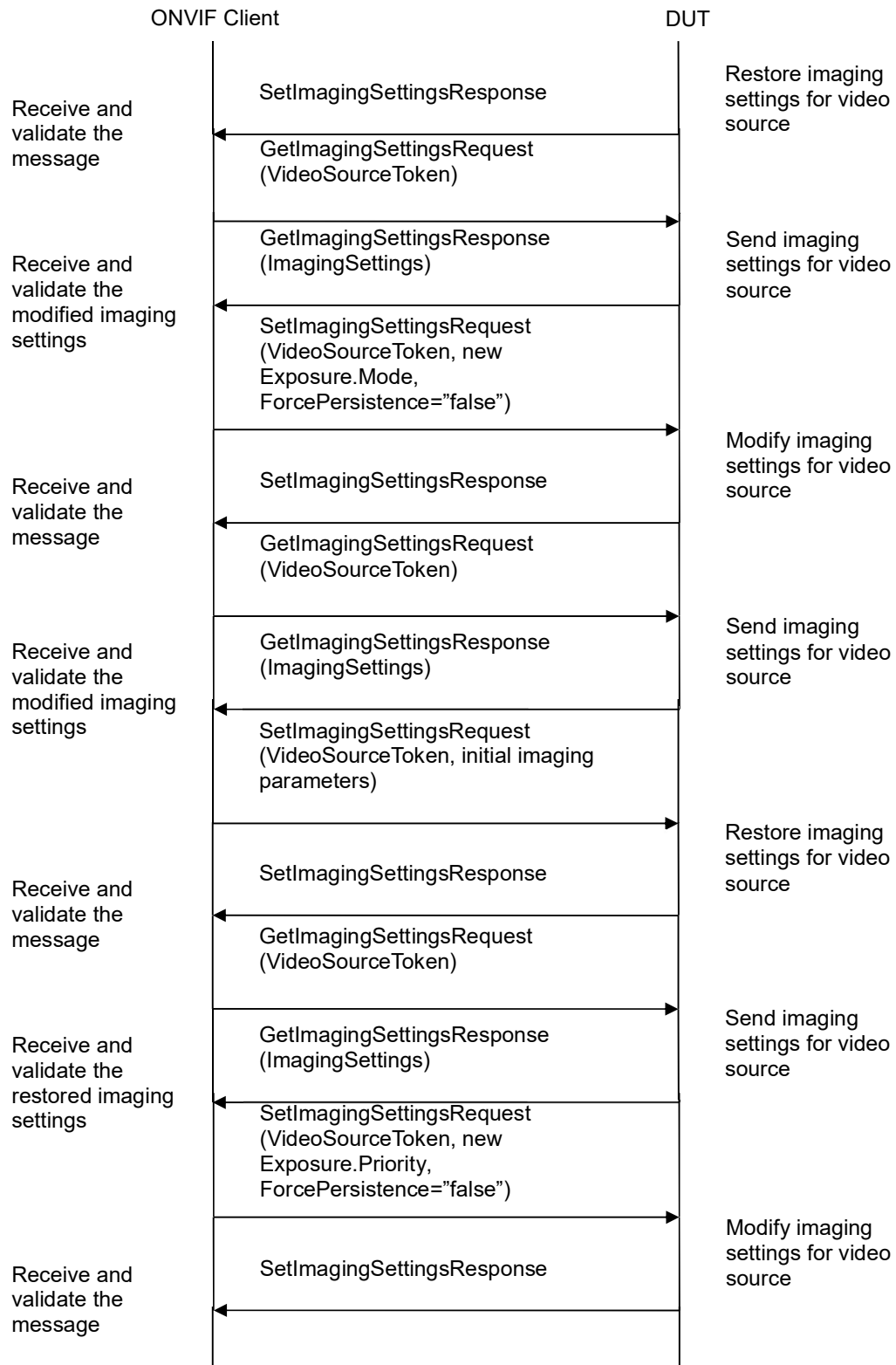
Test Sequence:

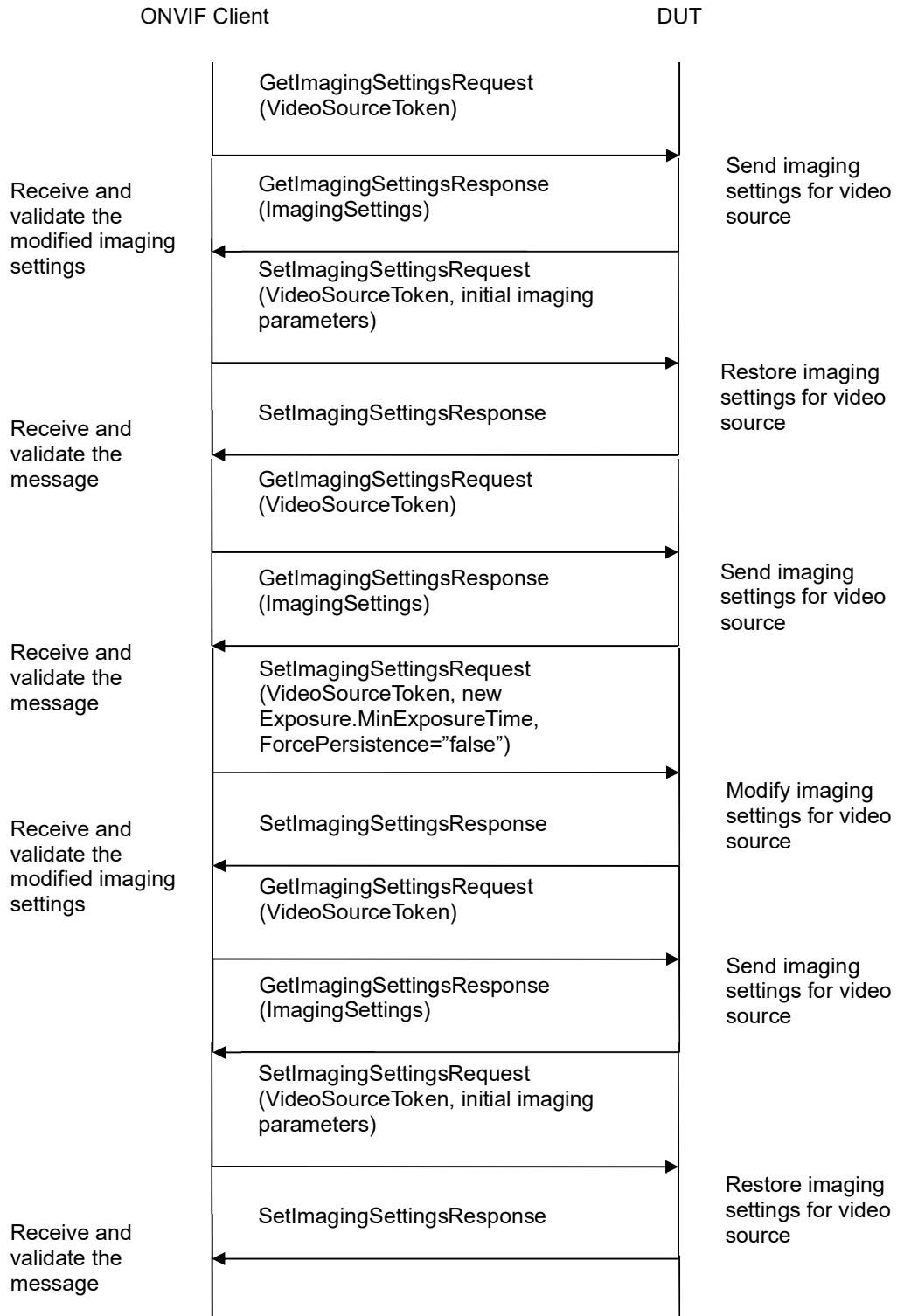


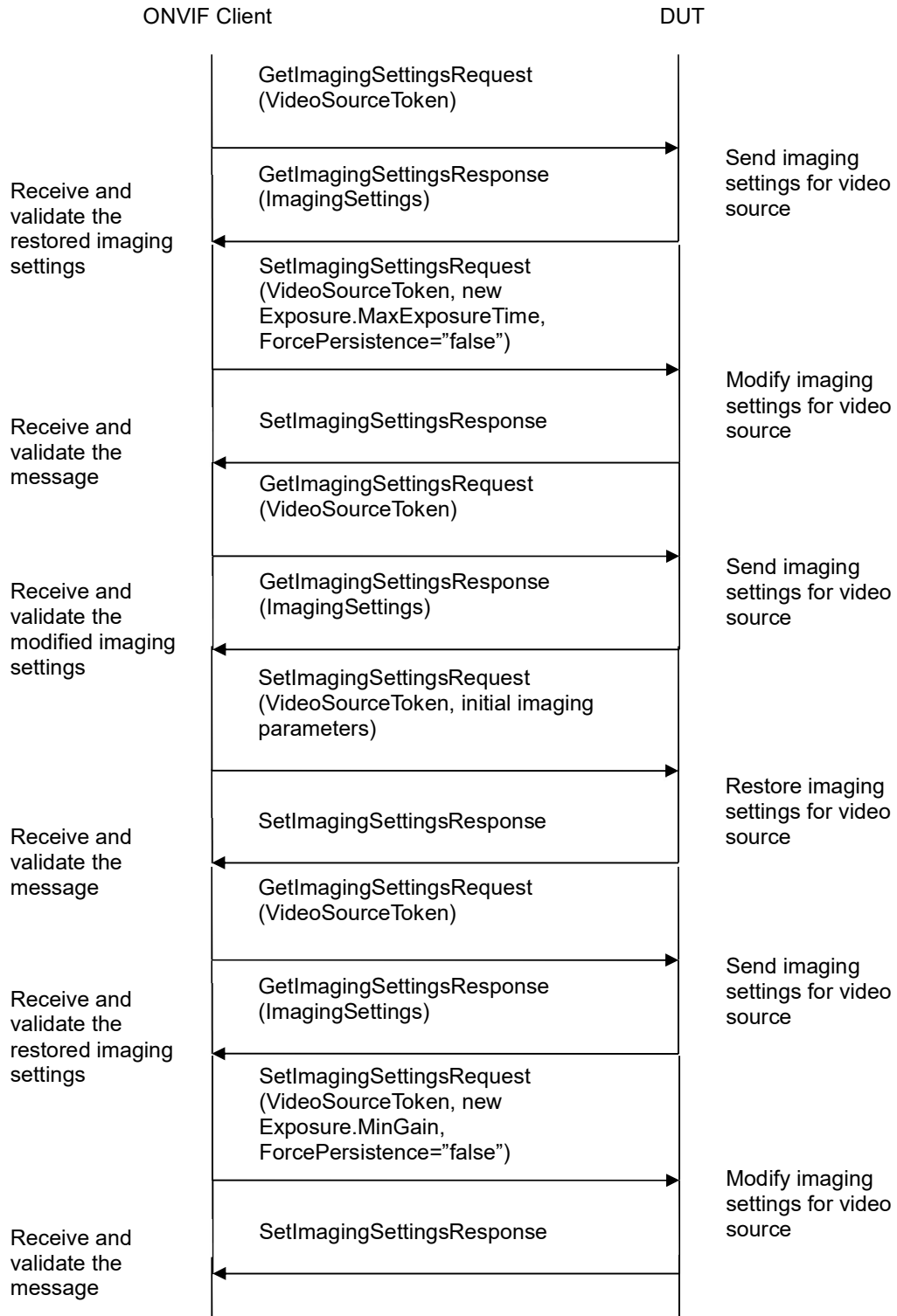


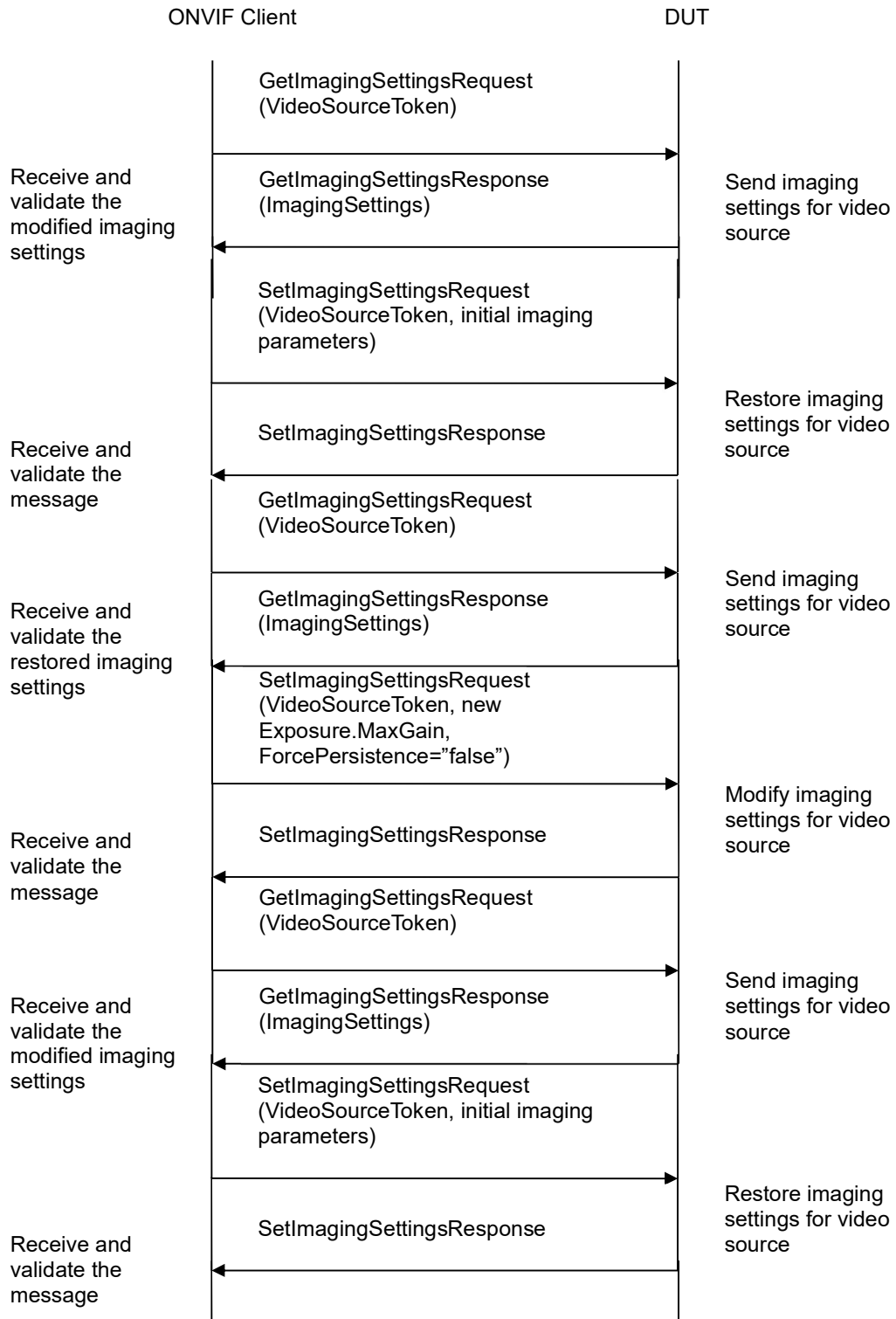


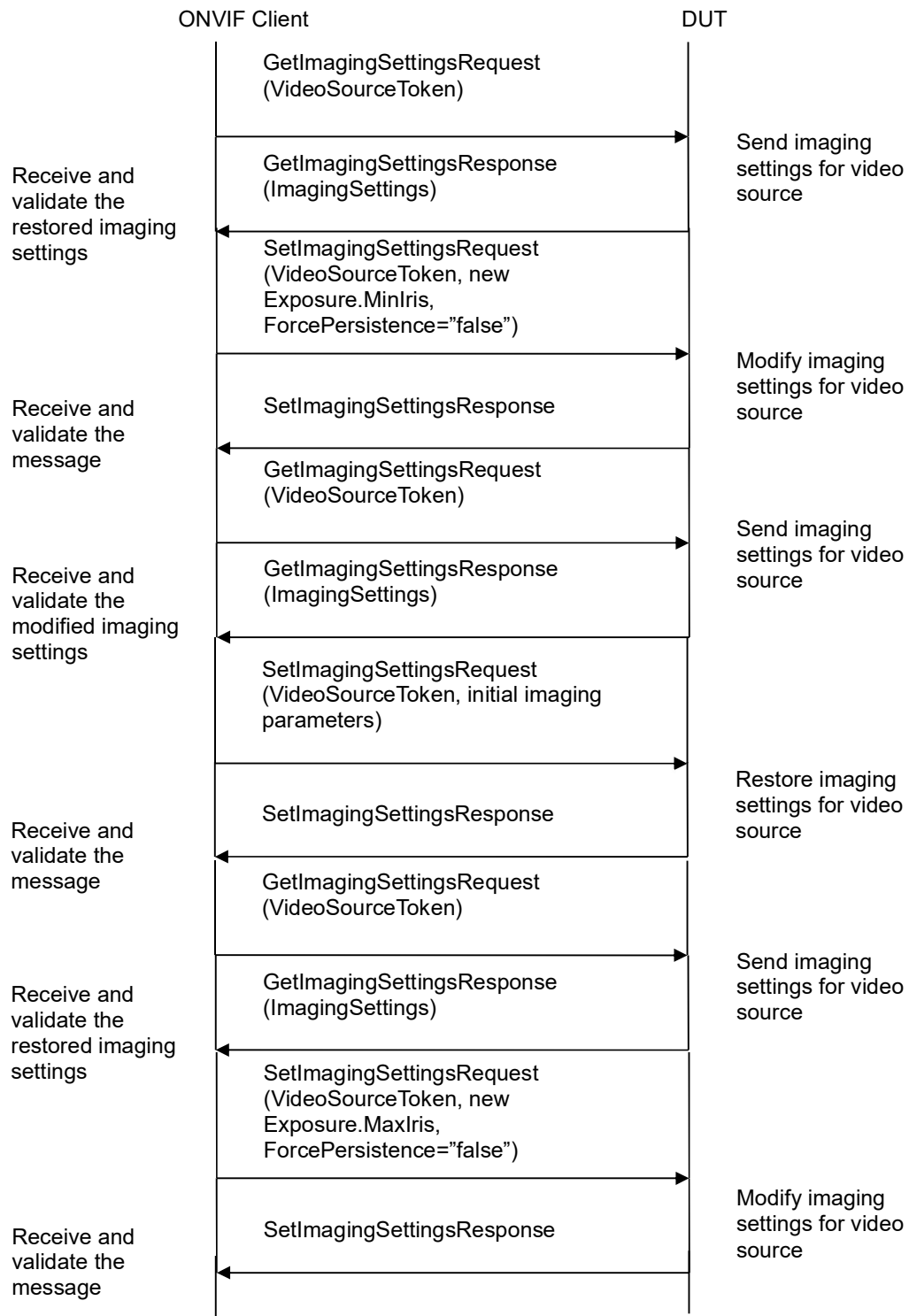


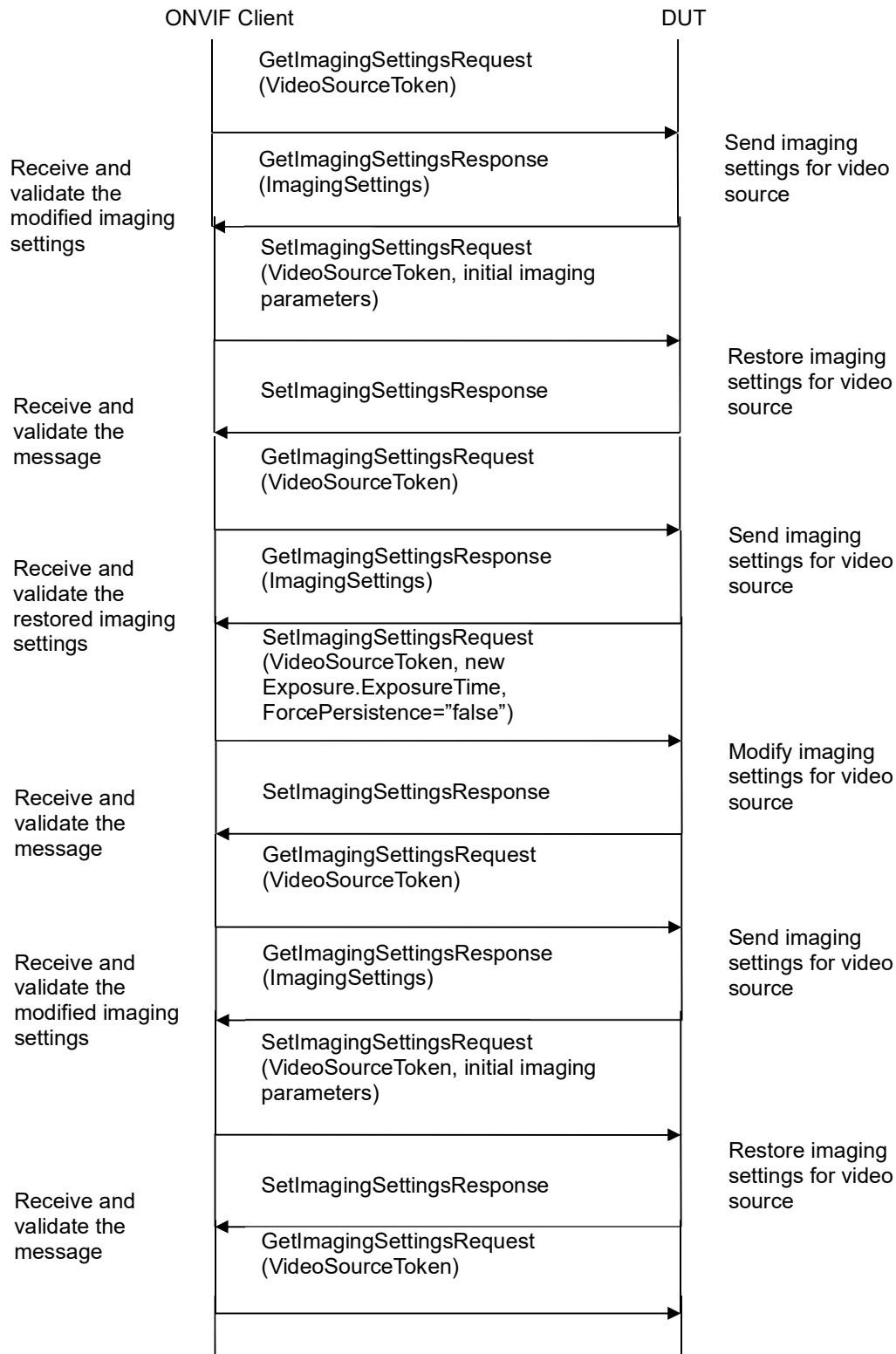


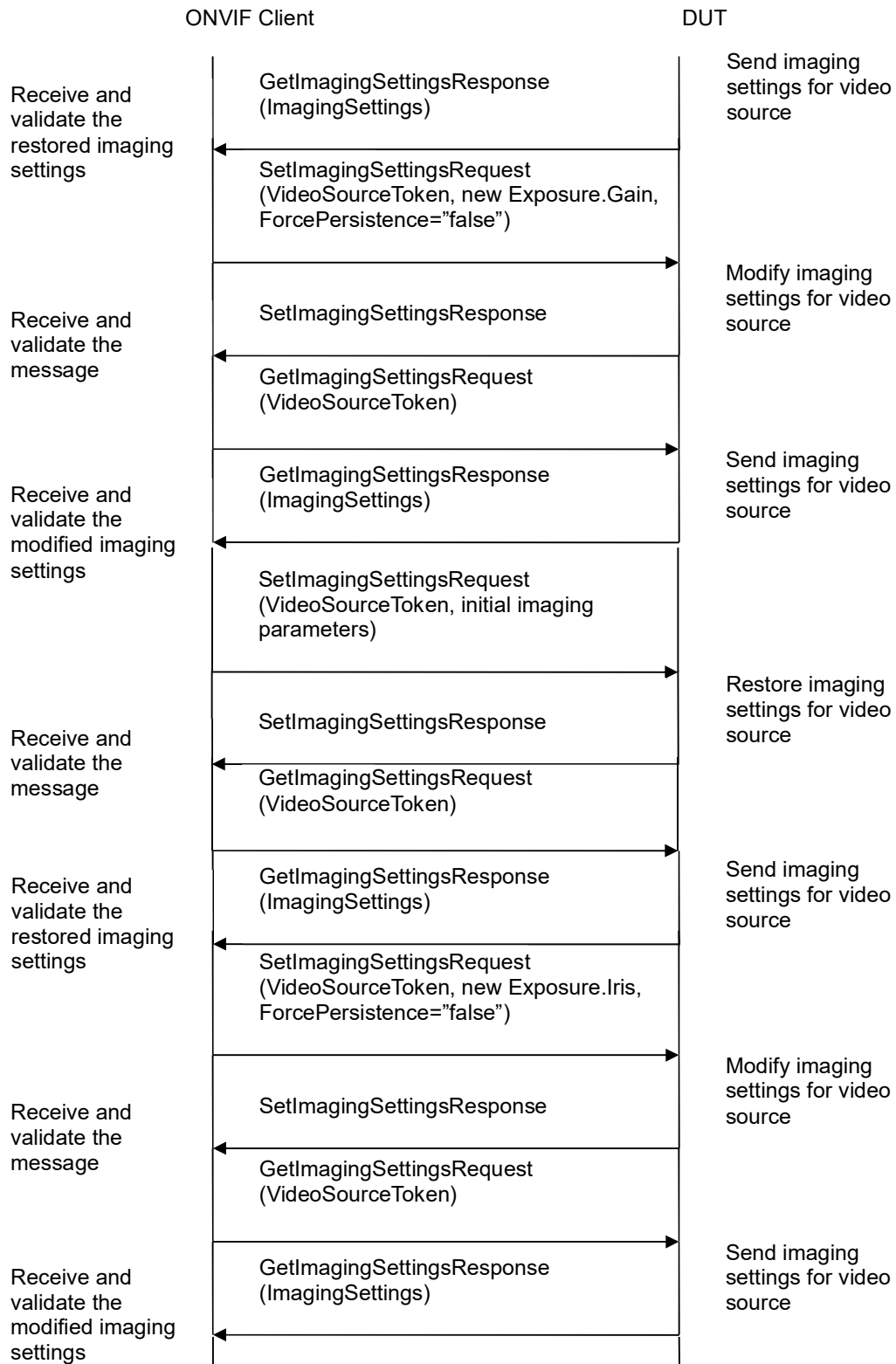


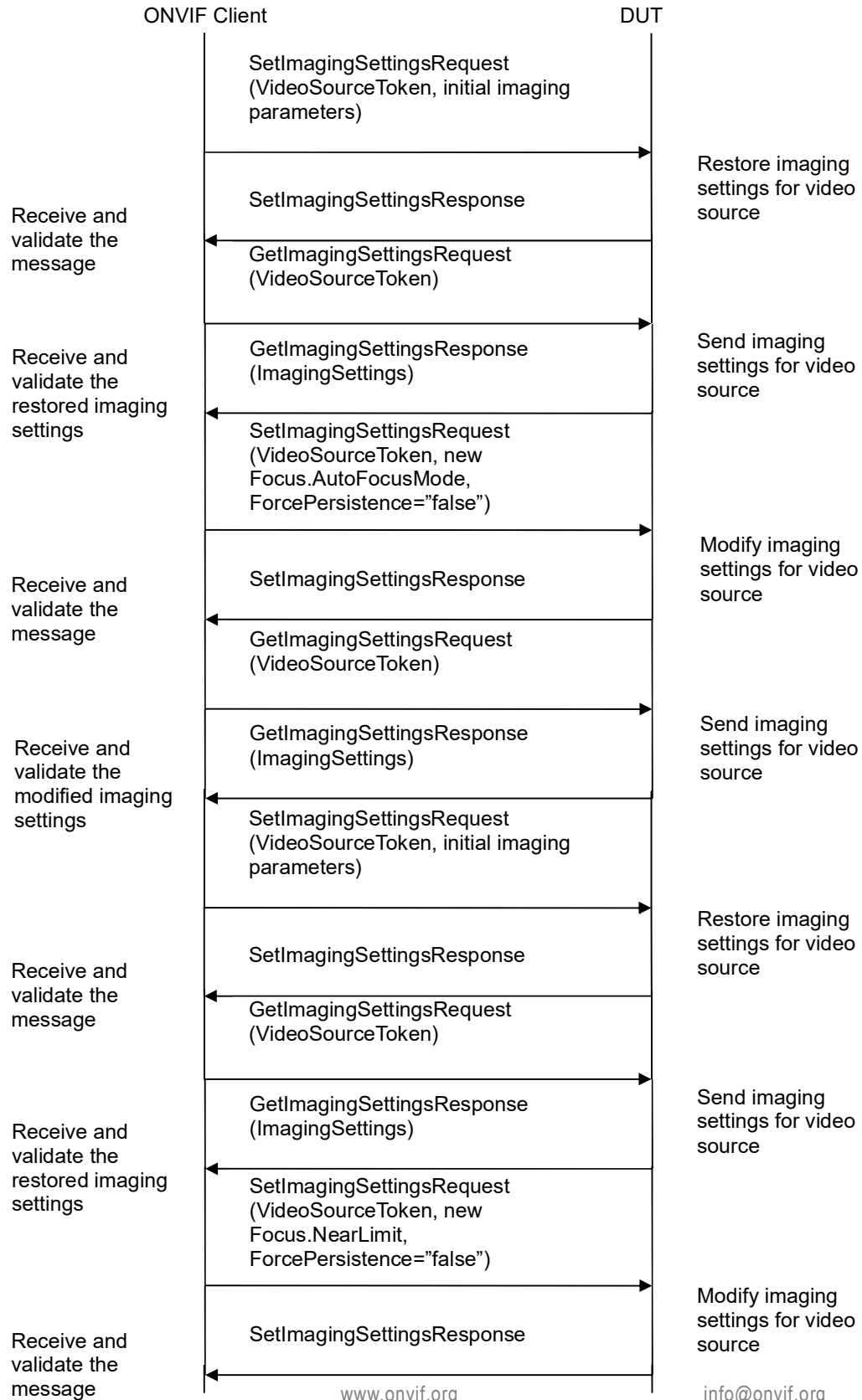


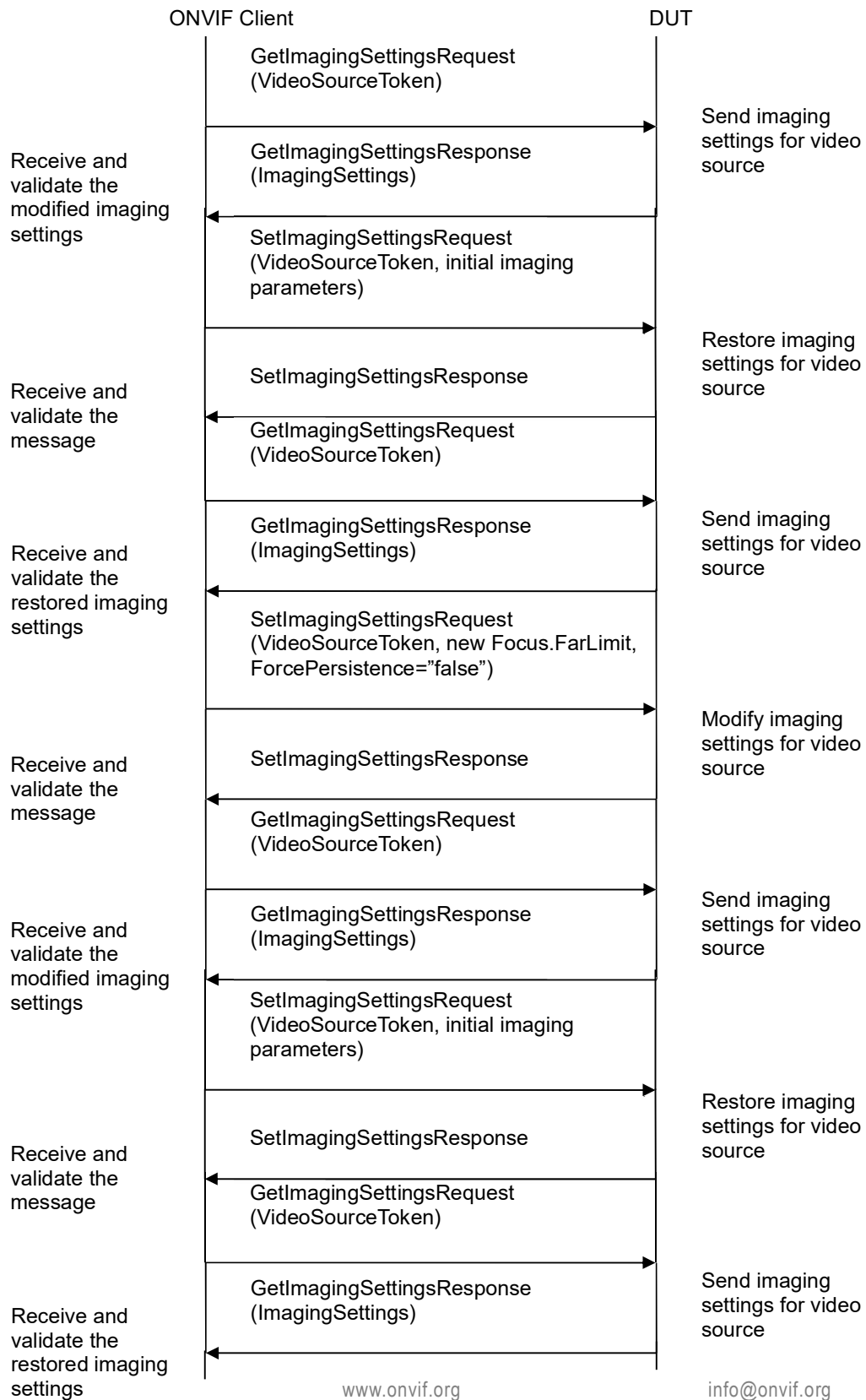


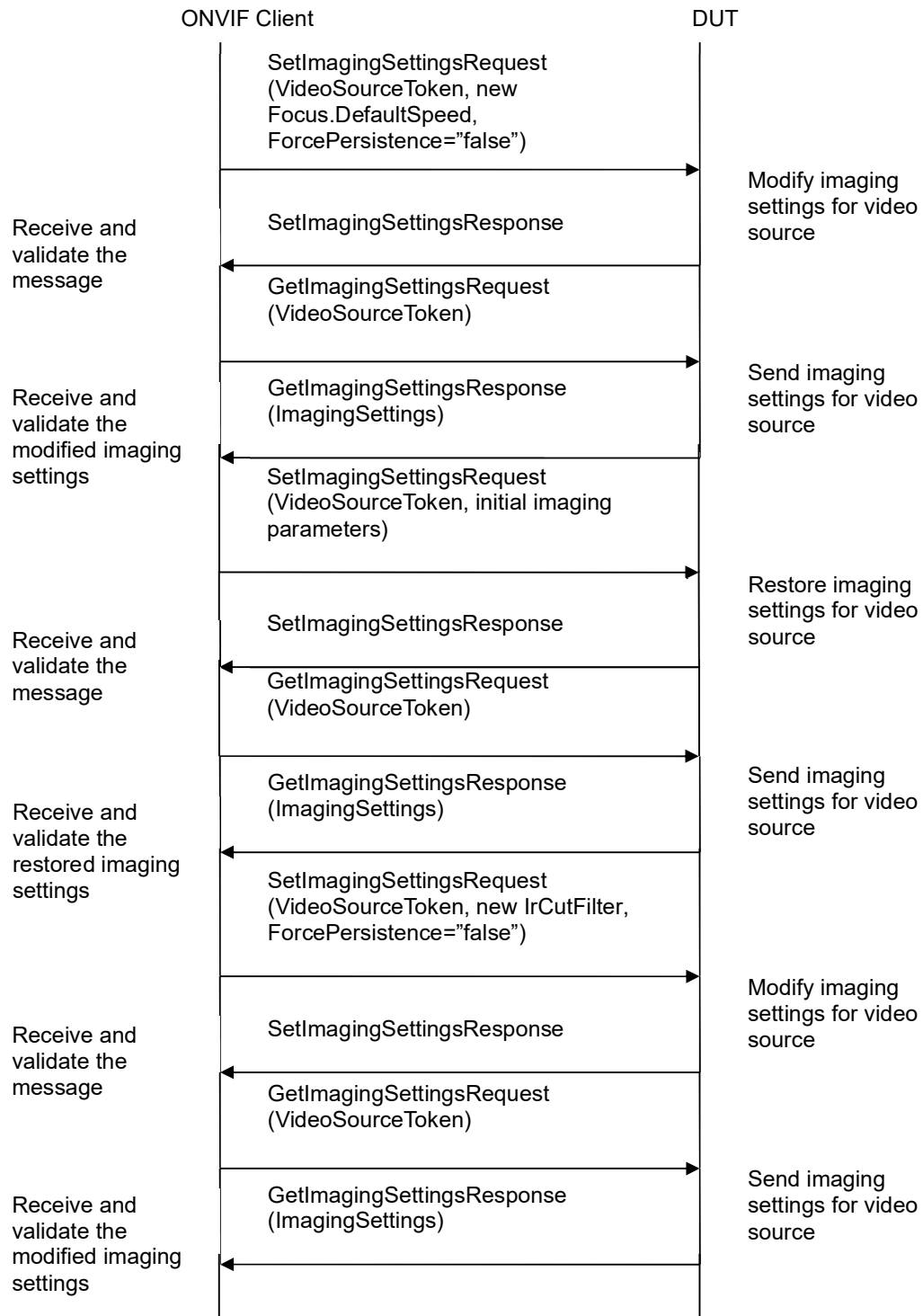


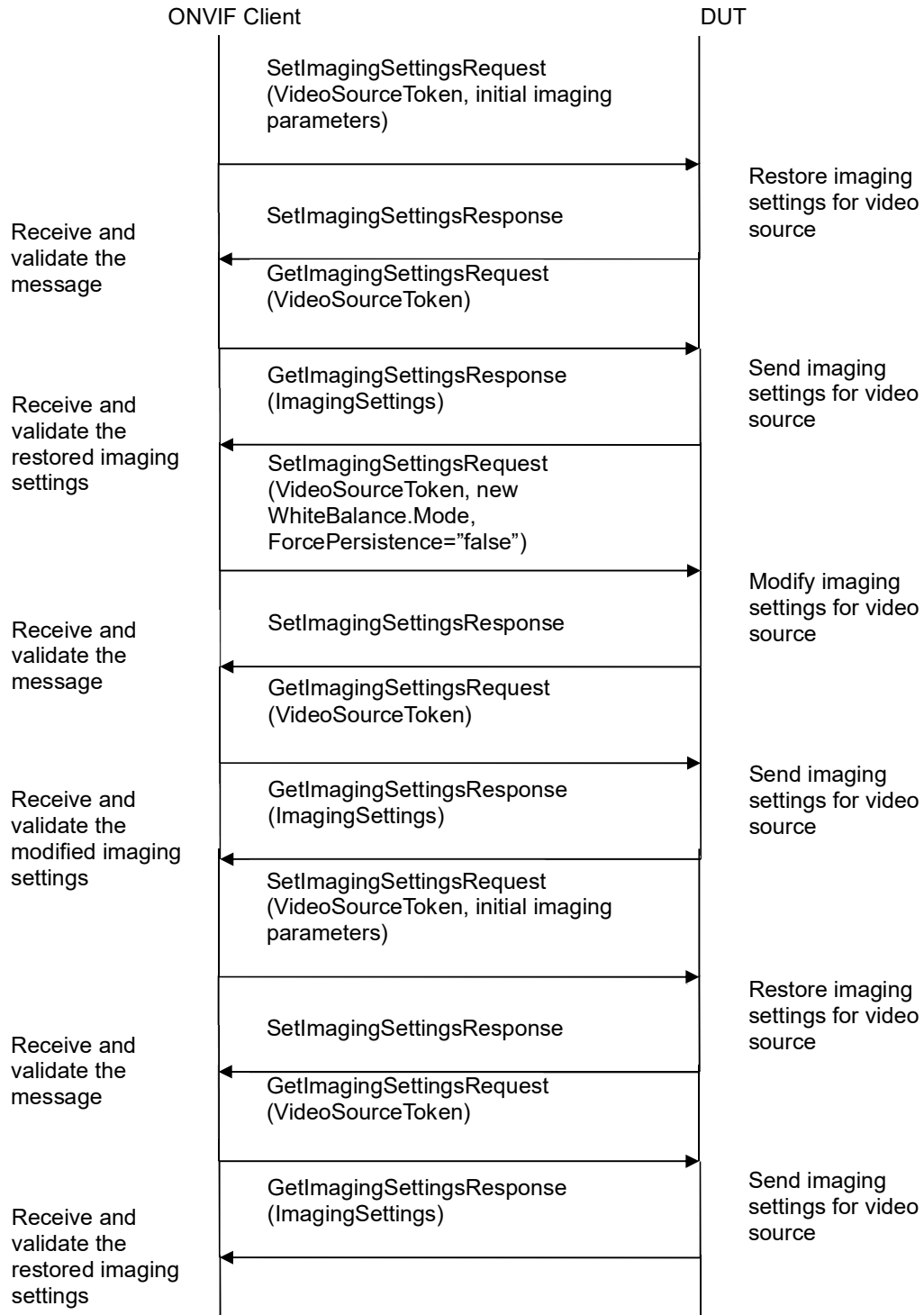


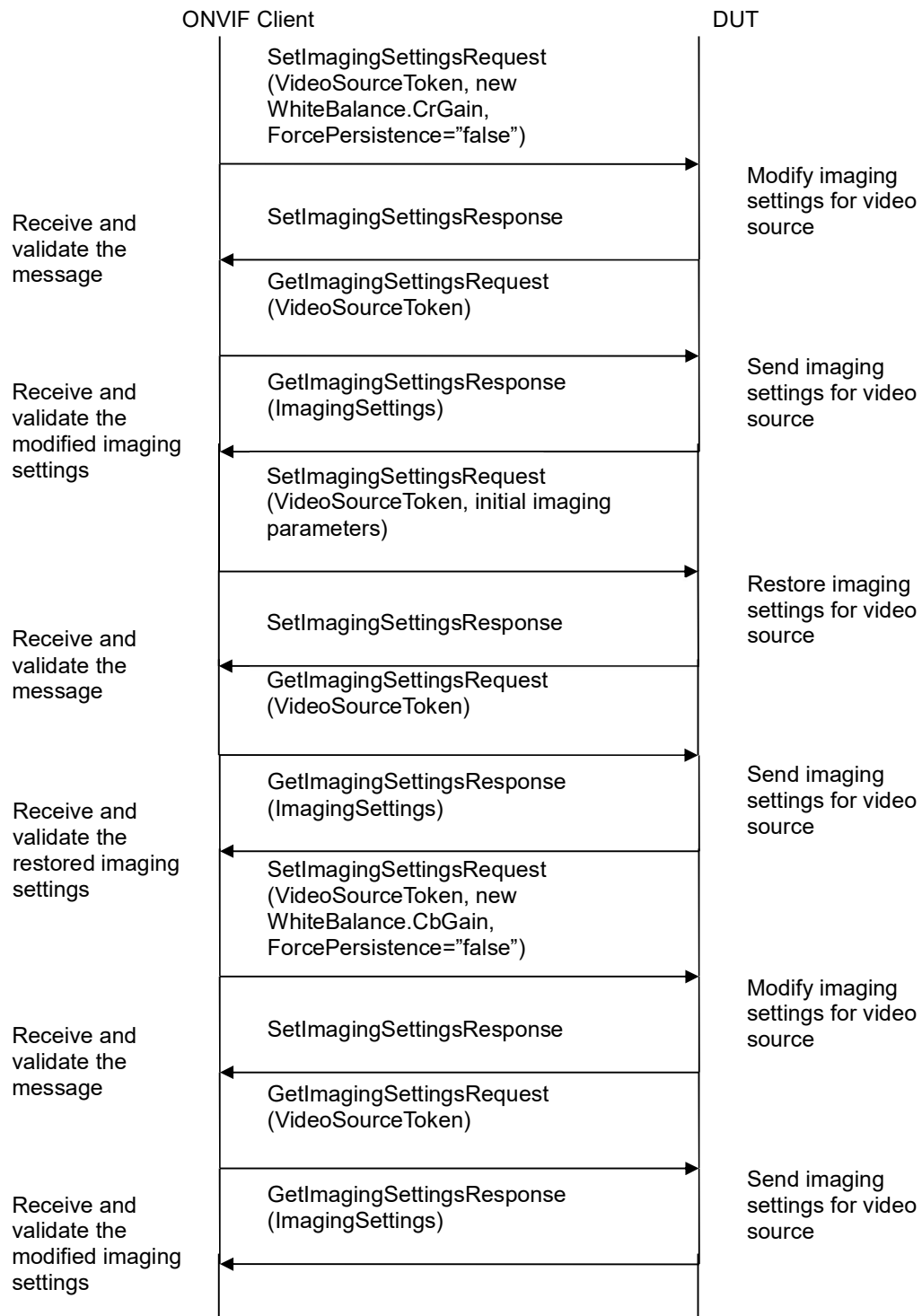


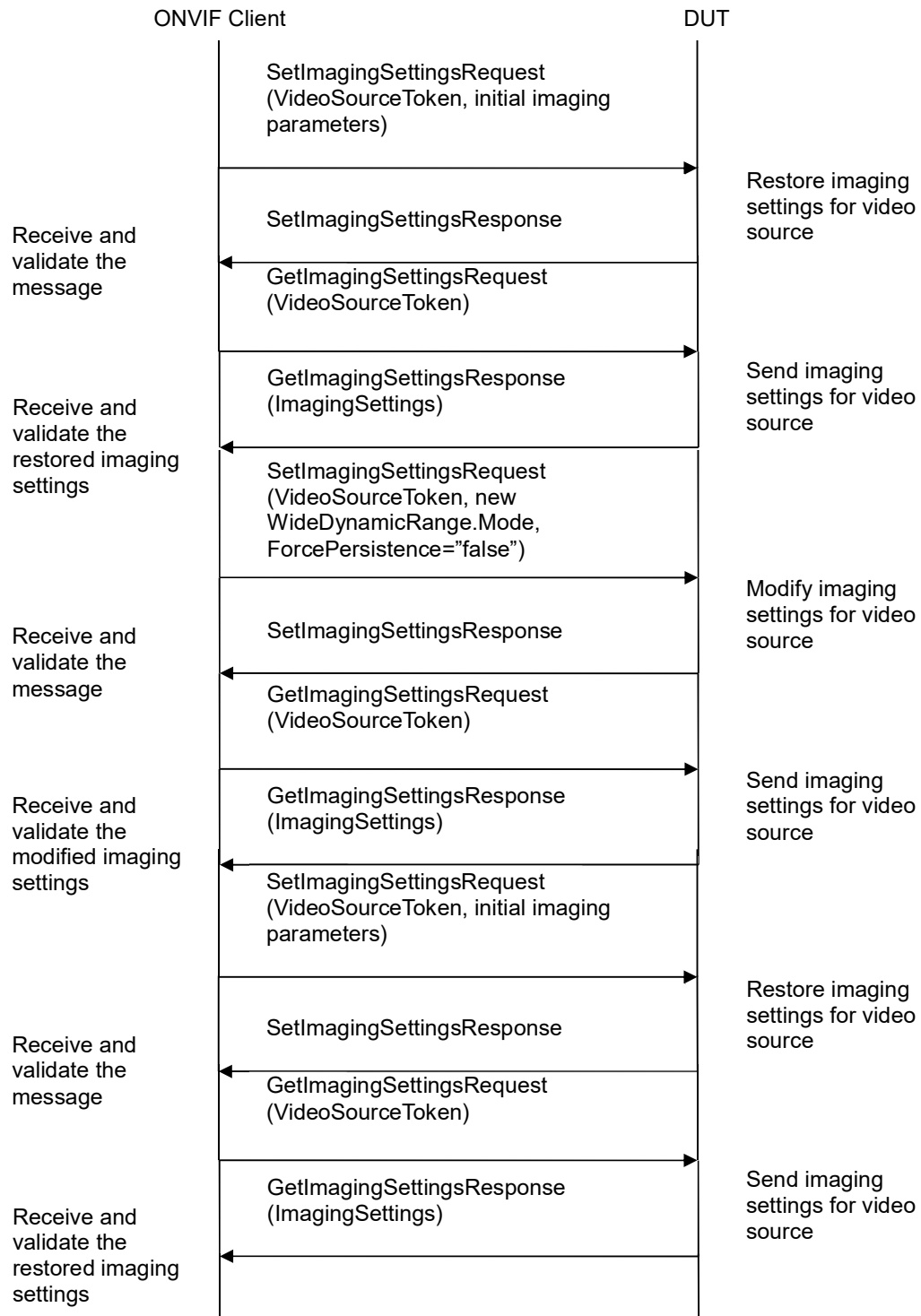


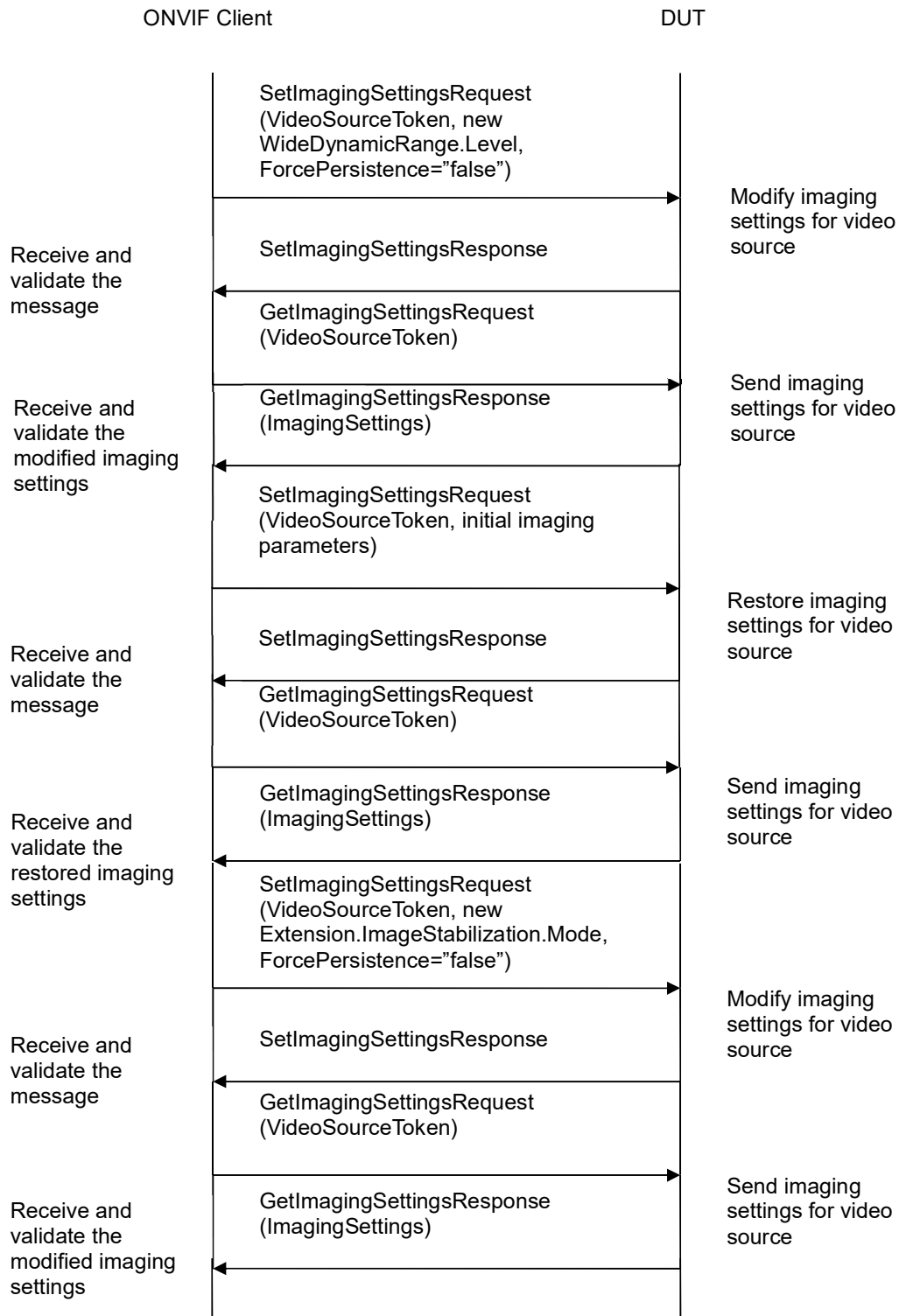


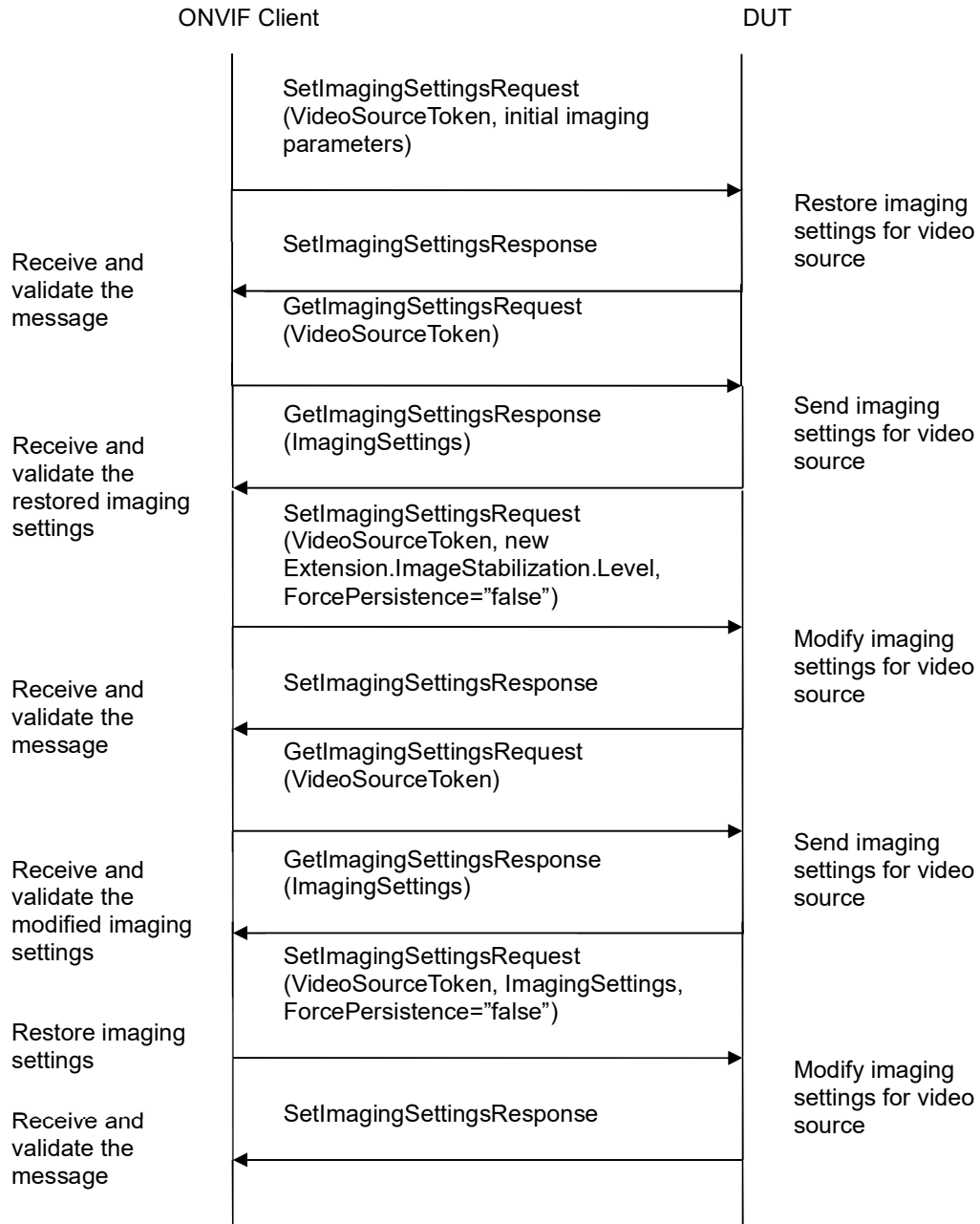












Test Procedure:

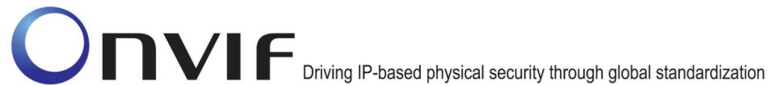
1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetOptionsRequest message (VideoSourceToken) to retrieve current imaging options for Video Source.



4. Verify the **GetOptionsResponse** message from the DUT.
5. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
6. Verify the **GetImagingSettingsResponse** message from the DUT.
7. If the DUT did not return BacklightCompensation setting in the **GetOptionsResponse** message, then go to the step 26.
8. If the DUT did not return two **BacklightCompensation.Mode** settings (ON and OFF) in the **GetOptionsResponse** message, then go to the step 17.
9. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, BacklightCompensation.Mode = backlightCompensationMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
10. Verify the **SetImagingSettingsResponse** message from the DUT.
11. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
12. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that BacklightCompensation.Mode setting was applied.
13. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
14. Verify the **SetImagingSettingsResponse** message from the DUT.
15. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
16. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
17. If DUT did not return **BacklightCompensation.Level** settings with Max > Min or BacklightCompensation.Mode = ON setting in the **GetOptionsResponse** message, then go to the step 26.
18. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, BacklightCompensation.Mode = ON, BacklightCompensation.Level = backlightCompensationLevel1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
19. Verify the **SetImagingSettingsResponse** message from the DUT.
20. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
21. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that BacklightCompensation.Level setting was applied.
22. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.



23. Verify the **SetImagingSettingsResponse** message from the DUT.
24. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
25. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
26. If the DUT did not return Brightness settings with Max > Min in the **GetOptionsResponse** message, then go to the step 35.
27. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Brightness = brightness1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
28. Verify the **SetImagingSettingsResponse** message from the DUT.
29. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
30. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Brightness setting was applied.
31. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
32. Verify the **SetImagingSettingsResponse** message from the DUT.
33. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
34. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
35. If the DUT did not return ColorSaturation settings with Max > Min in the **GetOptionsResponse** message, then go to the step 44.
36. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, ColorSaturation = colorSaturation1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
37. Verify the **SetImagingSettingsResponse** message from the DUT.
38. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
39. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that ColorSaturation setting was applied.
40. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
41. Verify the **SetImagingSettingsResponse** message from the DUT.
42. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.



43. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
44. If the DUT did not return Contrast settings with Max > Min in the **GetOptionsResponse** message then go to the step 53.
45. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Contrast = contrast1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
46. Verify the **SetImagingSettingsResponse** message from the DUT.
47. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
48. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Contrast setting was applied.
49. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
50. Verify the **SetImagingSettingsResponse** message from the DUT.
51. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
52. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
53. If the DUT did not return Sharpness settings with Max > Min in the **GetOptionsResponse** message, then go to the step 62.
54. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Sharpness = sharpness1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
55. Verify the **SetImagingSettingsResponse** message from the DUT.
56. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
57. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Sharpness setting was applied.
58. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
59. Verify the **SetImagingSettingsResponse** message from the DUT.
60. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
61. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
62. If the DUT did not return Exposure setting in the **GetOptionsResponse** message, then go to the step 164.



63. If the DUT did not return two Exposure.Mode settings (AUTO and MANUAL) in the **GetOptionsResponse** message, then go to the step 72.
64. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = exposureMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
65. Verify the **SetImagingSettingsResponse** message from the DUT.
66. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
67. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.Mode setting was applied.
68. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
69. Verify the **SetImagingSettingsResponse** message from the DUT.
70. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
71. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
72. If the DUT did not return Exposure.Mode = AUTO setting in the **GetOptionsResponse** message, then go to the step 136.
73. If the DUT did not return Exposure.Priority settings with Max > Min in the **GetOptionsResponse** message, then go to the step 82.
74. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = AUTO, Exposure.Priority = exposurePriority1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
75. Verify the **SetImagingSettingsResponse** message from the DUT.
76. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
77. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.Priority settings were applied.
78. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
79. Verify the **SetImagingSettingsResponse** message from the DUT.
80. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
81. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.



82. If the DUT did not return `Exposure.MinExposureTime` settings with `Max > Min` in the **GetOptionsResponse** message, then go to the step 91.
83. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `Exposure.Mode = AUTO`, `Exposure.MinExposureTime = exposureMinExposureTime1` (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
84. Verify the **SetImagingSettingsResponse** message from the DUT.
85. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
86. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that `Exposure.MinExposureTime` setting was applied.
87. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
88. Verify the **SetImagingSettingsResponse** message from the DUT.
89. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
90. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
91. If the DUT did not return `Exposure.MaxExposureTime` settings with `Max > Min` in the **GetOptionsResponse** message, then go to the step 100.
92. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `Exposure.Mode = AUTO`, `Exposure.MaxExposureTime = exposureMaxExposureTime1` (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
93. Verify the **SetImagingSettingsResponse** message from the DUT.
94. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
95. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that `Exposure.MaxExposureTime` setting was applied.
96. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
97. Verify the **SetImagingSettingsResponse** message from the DUT.
98. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
99. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
100. If the DUT did not return `Exposure.MinGain` settings with `Max > Min` in the **GetOptionsResponse** message, then go to the step 109.



101. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = AUTO, Exposure.MinGain = exposureMinGain1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
102. Verify the **SetImagingSettingsResponse** message from the DUT.
103. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
104. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.MinGain setting was applied.
105. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
106. Verify the **SetImagingSettingsResponse** message from the DUT.
107. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
108. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
109. If the DUT did not return Exposure.MaxGain settings with Max > Min in the **GetOptionsResponse** message, then go to the step 118.
110. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = AUTO, Exposure.MaxGain = exposureMaxGain1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
111. Verify the **SetImagingSettingsResponse** message from the DUT.
112. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
113. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.MaxGain setting was applied.
114. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
115. Verify the **SetImagingSettingsResponse** message from the DUT.
116. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
117. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
118. If the DUT did not return Exposure.MinIris settings with Max > Min in the **GetOptionsResponse** message, then go to the step 127.
119. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = AUTO, Exposure.MinIris = exposureMinIris1 (other than current), other



- imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
120. Verify the **SetImagingSettingsResponse** message from the DUT.
 121. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 122. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.MinIris setting was applied.
 123. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
 124. Verify the **SetImagingSettingsResponse** message from the DUT.
 125. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 126. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
 127. If the DUT did not return Exposure.MaxIris settings with Max > Min in the **GetOptionsResponse** message, then go to the step 136.
 128. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = AUTO, Exposure.MaxIris = exposureMaxIris1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
 129. Verify the **SetImagingSettingsResponse** message from the DUT.
 130. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 131. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.MaxIris setting was applied.
 132. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
 133. Verify the **SetImagingSettingsResponse** message from the DUT.
 134. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 135. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
 136. If the DUT did not return Exposure.Mode = MANUAL setting in the **GetOptionsResponse** message, then go to the step 164.
 137. If the DUT did not return Exposure.ExposureTime settings with Max > Min in the **GetOptionsResponse** message, then go to the step 146.
 138. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = MANUAL, Exposure.ExposureTime = exposureExposureTime1 (other than



- current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
139. Verify the **SetImagingSettingsResponse** message from the DUT.
 140. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 141. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.ExposureTime settings were applied.
 142. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
 143. Verify the **SetImagingSettingsResponse** message from the DUT.
 144. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 145. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
 146. If the DUT did not return Exposure.Gain settings with Max > Min in the **GetOptionsResponse** message, then go to the step 155.
 147. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = MANUAL, Exposure.Gain = exposureGain1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
 148. Verify the **SetImagingSettingsResponse** message from the DUT.
 149. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 150. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.Gain settings were applied.
 151. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
 152. Verify the **SetImagingSettingsResponse** message from the DUT.
 153. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
 154. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
 155. If the DUT did not return Exposure.Iris settings with Max > Min in the **GetOptionsResponse** message, then go to the step 164.
 156. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Exposure.Mode = MANUAL, Exposure.Iris = exposureIris1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
 157. Verify the **SetImagingSettingsResponse** message from the DUT.



158. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
159. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Exposure.Iris settings were applied.
160. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
161. Verify the **SetImagingSettingsResponse** message from the DUT.
162. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
163. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
164. If the DUT did not return Focus setting in the **GetOptionsResponse** message, then go to the step 202.
165. If the DUT did not return at least two Focus.AutoFocusModes settings (AUTO or MANUAL) in the **GetOptionsResponse** message, then go to the step 174.
166. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Focus.AutoFocusMode = focusAutoFocusMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
167. Verify the **SetImagingSettingsResponse** message from the DUT.
168. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
169. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Focus.AutoFocusMode setting was applied.
170. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
171. Verify the **SetImagingSettingsResponse** message from the DUT.
172. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
173. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
174. If the DUT did not return Focus.AutoFocusModes = AUTO setting in the **GetOptionsResponse** message, then go to the step 193.
175. If the DUT did not return Focus.NearLimit settings with Max > Min in the **GetOptionsResponse** message, then go to the step 184.
176. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Focus.AutoFocusModes = AUTO, Focus.NearLimit = focusNearLimit1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.



177. Verify the **SetImagingSettingsResponse** message from the DUT.
178. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
179. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Focus.NearLimit setting was applied.
180. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
181. Verify the **SetImagingSettingsResponse** message from the DUT.
182. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
183. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
184. If the DUT did not return Focus.FarLimit settings with Max > Min in the **GetOptionsResponse** message, then go to the step 193.
185. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Focus.AutoFocusModes = AUTO, Focus.FarLimit = focusFarLimit1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
186. Verify the **SetImagingSettingsResponse** message from the DUT.
187. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
188. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Focus.FarLimit setting was applied.
189. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
190. Verify the **SetImagingSettingsResponse** message from the DUT.
191. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
192. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
193. If the DUT did not return Focus.AutoFocusModes = MANUAL setting or Focus.DefaultSpeed settings with Max > Min in the **GetOptionsResponse** message, then go to the step 202.
194. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Focus.AutoFocusModes = MANUAL, Focus.DefaultSpeed = focusDefaultSpeed1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
195. Verify the **SetImagingSettingsResponse** message from the DUT.



196. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
197. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Focus.DefaultSpeed setting was applied.
198. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
199. Verify the **SetImagingSettingsResponse** message from the DUT.
200. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
201. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
202. If the DUT did not return at least two IrCutFilterModes settings in the **GetOptionsResponse** message, then go to the step 211.
203. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, IrCutFilter = irCutFilter1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
204. Verify the **SetImagingSettingsResponse** message from the DUT.
205. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
206. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that IrCutFilter setting was applied.
207. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
208. Verify the **SetImagingSettingsResponse** message from the DUT.
209. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging **settings**.
210. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
211. If the DUT did not return WhiteBalance setting in the **GetOptionsResponse** message, then go to the step 240.
212. If the DUT did not return at least two WhiteBalance.Mode settings (AUTO or MANUAL) in the **GetOptionsResponse** message, then go to the step 221.
213. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, WhiteBalance.Mode = whiteBalanceMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
214. Verify the **SetImagingSettingsResponse** message from the DUT.
215. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.



216. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that WhiteBalance.Mode setting was applied.
217. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
218. Verify the **SetImagingSettingsResponse** message from the DUT.
219. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
220. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
221. If the DUT did not return WhiteBalance.Mode = MANUAL setting in the **GetOptionsResponse** message, then go to the step 240.
222. If DUT did not return WhiteBalance.YrGain settings with Max > Min in the **GetOptionsResponse** message, then go to the step 231.
223. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, WhiteBalance.Mode = MANUAL, WhiteBalance.CrGain = whiteBalanceCrGain1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
224. Verify the **SetImagingSettingsResponse** message from the DUT.
225. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
226. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that WhiteBalance.CrGain setting was applied.
227. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
228. Verify the **SetImagingSettingsResponse** message from the DUT.
229. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
230. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
231. If the DUT did not return WhiteBalance.YbGain settings with Max > Min in the **GetOptionsResponse** message, then go to the step 240.
232. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, WhiteBalance.Mode = MANUAL, WhiteBalance.CbGain = whiteBalanceCbGain1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
233. Verify the **SetImagingSettingsResponse** message from the DUT.
234. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.



235. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that WhiteBalance.CbGain setting was applied.
236. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
237. Verify the **SetImagingSettingsResponse** message from the DUT.
238. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
239. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
240. If the DUT did not return WideDynamicRange setting in the **GetOptionsResponse** message, then go to the step 259.
241. If the DUT did not return two WideDynamicRange.Mode settings (ON and OFF) in the **GetOptionsResponse** message, then go to the step 250.
242. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, WideDynamicRange.Mode = wideDynamicRangeMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
243. Verify the **SetImagingSettingsResponse** message from the DUT.
244. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
245. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that WideDynamicRange.Mode setting was applied.
246. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
247. Verify the **SetImagingSettingsResponse** message from the DUT.
248. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
249. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
250. If the DUT did not return WideDynamicRange.Level settings with Max > Min or WideDynamicRange.Mode = ON setting in the **GetOptionsResponse** message, then go to the step 259.
251. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, WideDynamicRange.Mode = ON, WideDynamicRange.Level = wideDynamicRangeLevel1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
252. Verify the **SetImagingSettingsResponse** message from the DUT.
253. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.



254. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that WideDynamicRange.Level setting was applied.
255. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
256. Verify the **SetImagingSettingsResponse** message from the DUT.
257. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
258. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
259. If the DUT did not return Extension.ImageStabilization settings in the **GetOptionsResponse** message, then go to the step 274.
260. If the DUT did not return at least two Extension.ImageStabilization.Mode settings in the **GetOptionsResponse** message, then go to the step 269.
261. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Extension.ImageStabilization.Mode = extensionImageStabilizationMode1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
262. Verify the **SetImagingSettingsResponse** message from the DUT.
263. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
264. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Extension.ImageStabilization.Mode setting was applied.
265. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to **initial** state.
266. Verify the SetImagingSettingsResponse message from the DUT.
267. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
268. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that changed settings were returned to initial state.
269. If the DUT did not return Extension.ImageStabilization.Level settings with Max > Min or Extension.ImageStabilization.Mode = ON setting in the **GetOptionsResponse** message, then go to the step 274.
270. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Extension.ImageStabilization.Mode = ON, Extension.ImageStabilization.Level = extensionImageStabilizationLevel1 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
271. Verify the **SetImagingSettingsResponse** message from the DUT.
272. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.



- 273. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that Extension.ImageStabilization.Level setting was applied.
- 274. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken) to restore old imaging settings.
- 275. Verify the **SetImagingSettingsResponse** message from the DUT.
- 276. ONVIF Client will repeat steps 3-275 for other video sources supported by the DUT.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **GetOptionsResponse** message.

The DUT did not send a valid **GetOptionsResponse** message.

The DUT did not send **SetImagingSettings** message.

The DUT did not send a valid **SetImagingSettings** message.

The DUT did not send **GetImagingSettings** message.

The DUT did not send a valid **GetImagingSettings** message.

The DUT did not modify imaging settings by invoked **SetImagingSettingsRequest** message.

The DUT did not return changed settings to initial state by invoked **SetImagingSettingsRequest** message.

Note: All imaging settings in each **SetImagingSettingsRequest** message shall have values from the range provided in **GetOptionsResponse** message. If options for imaging setting were skipped in **GetOptionsResponse** message, then they shall be skipped in **SetImagingSettingsRequest** message.

Note: At step 203 the ONVIF test tool selects irCutFilter1 = OFF if it is supported by the DUT and if it is not a current IrCutFilterMode.

4.1.8 IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL FEATURES

Test Label: Imaging DUT SetImagingSettings Command Validation.

Test Case ID: IMAGING-1-1-15

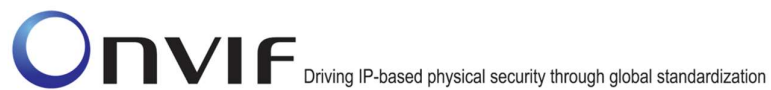
ONVIF Core Specification Coverage: Set imaging settings

Command under Test: SetImagingSettings

WSDL Reference: imaging.wsdl

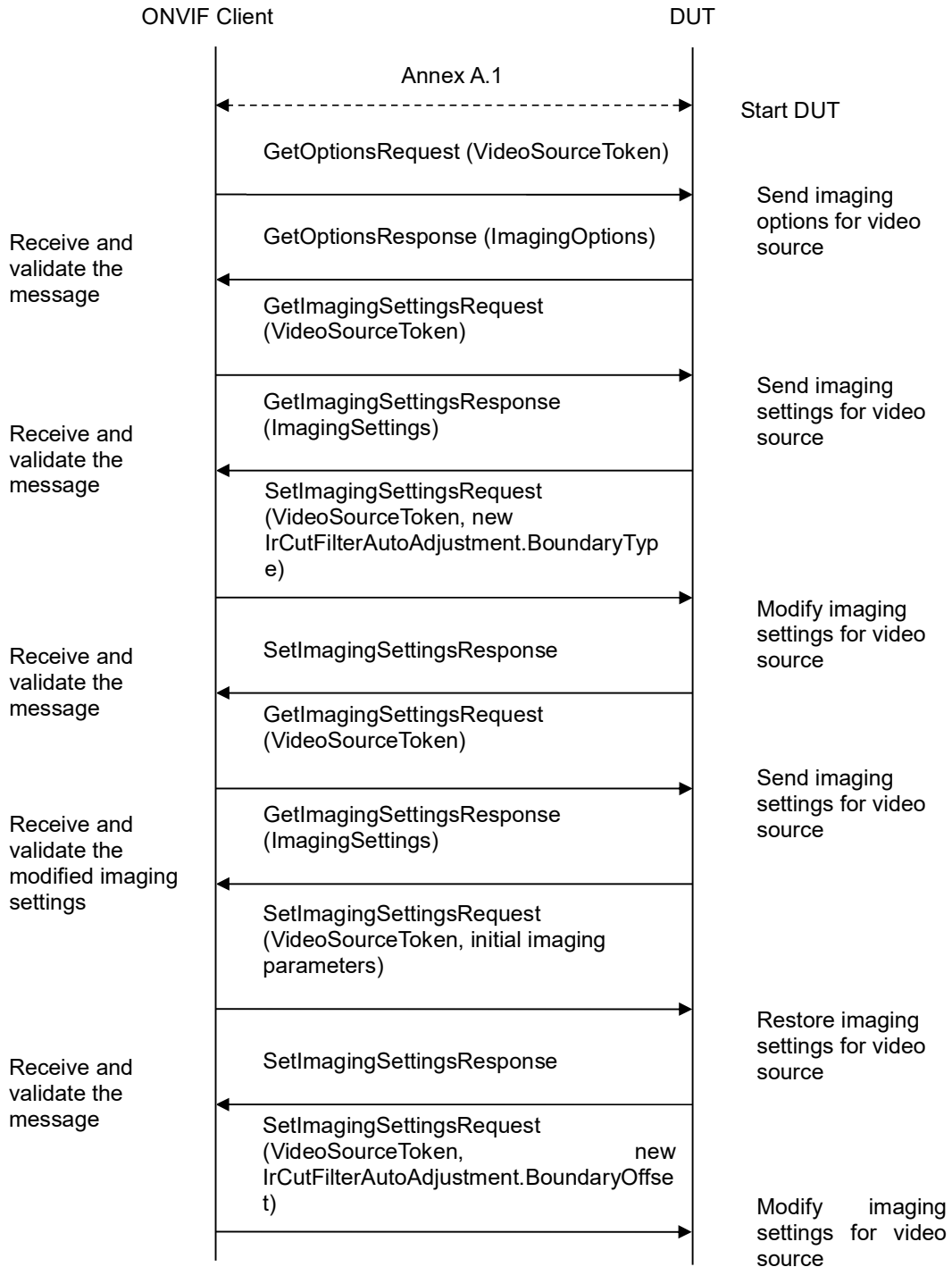
Test Propose: To verify the behaviour of DUT SetImagingSettings additional features.

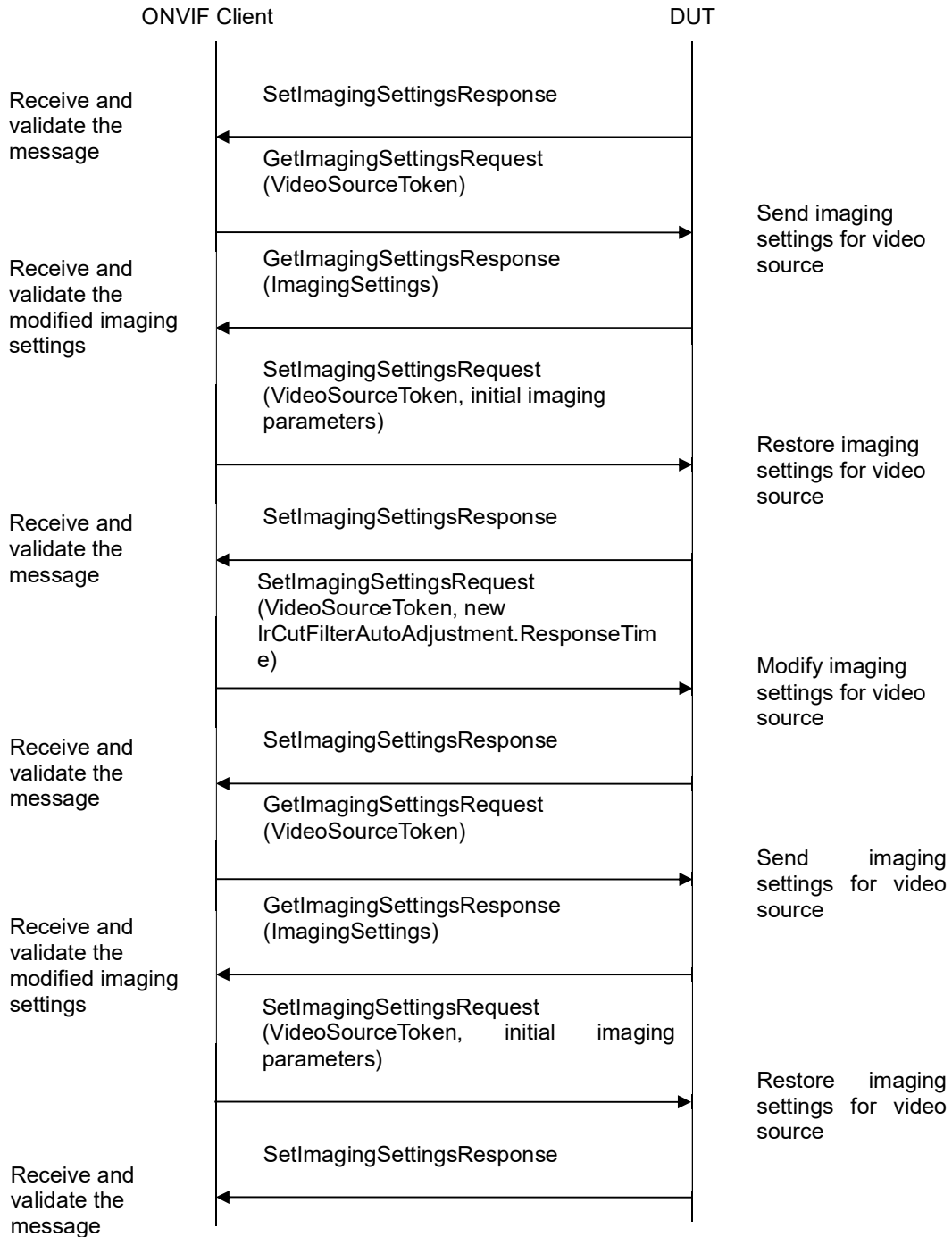
Pre-Requisite: None

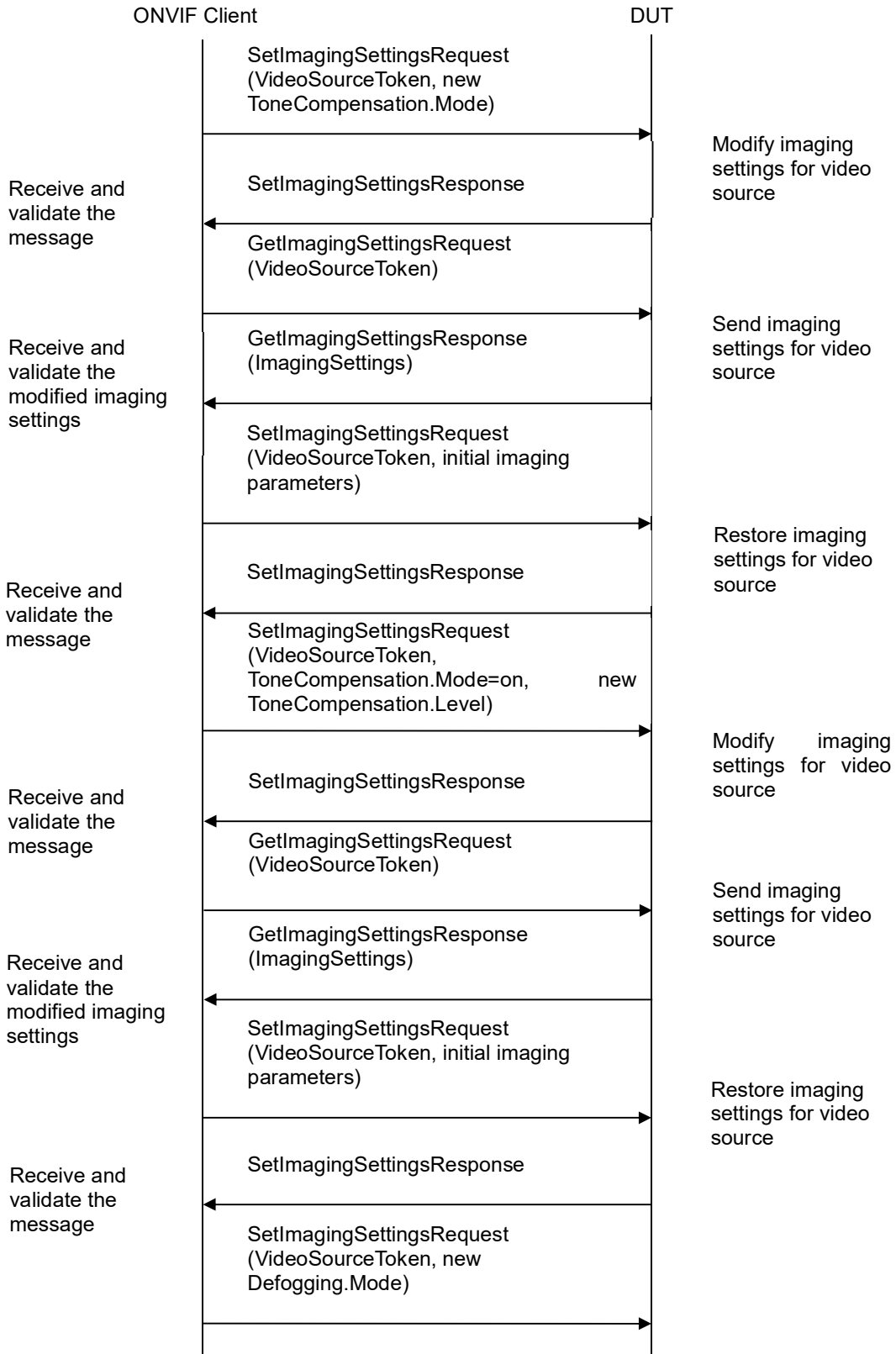


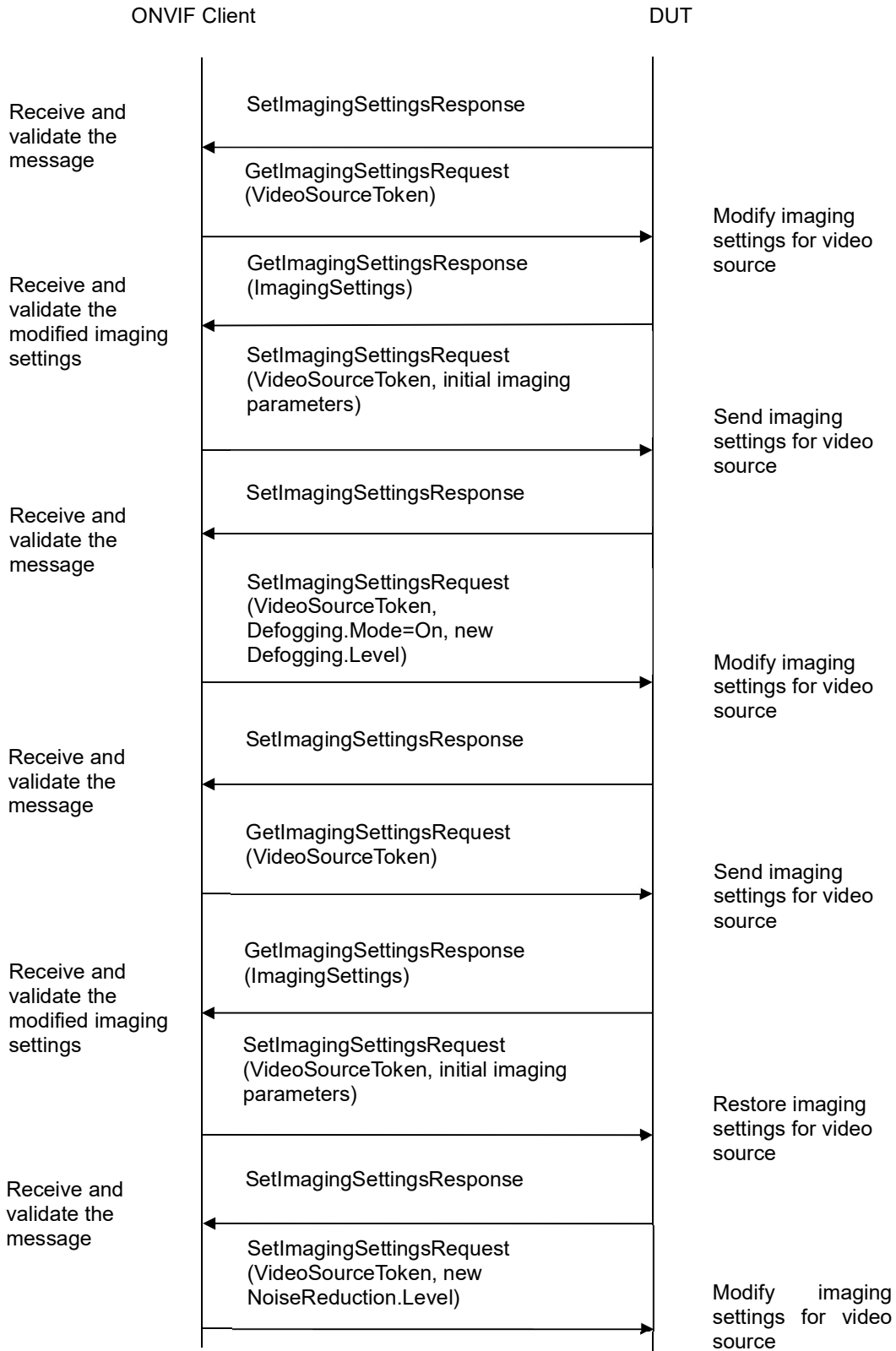
Test Configuration: ONVIF Client and DUT

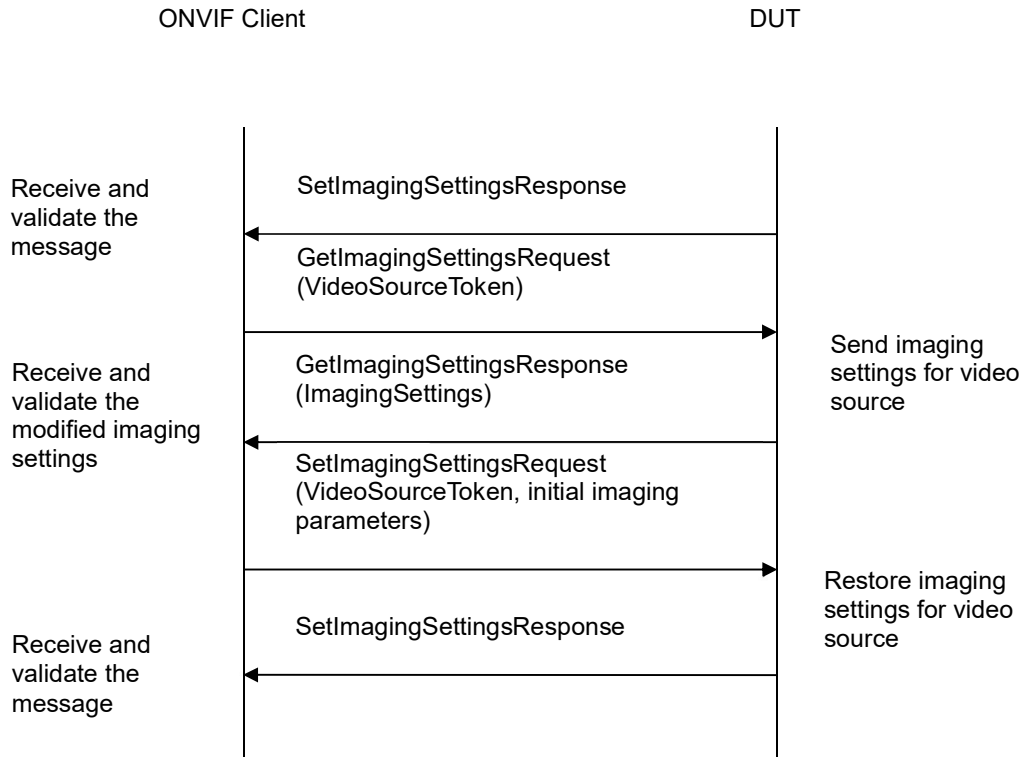
Test Sequence:











Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke **GetOptionsRequest** message (VideoSourceToken) to retrieve current imaging options for Video Source.
4. Verify the **GetOptionsResponse** message from the DUT.
5. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
6. Verify the **GetImagingSettingsResponse** message from the DUT.
7. If the DUT did not return at least two IrCutFilterAutoAdjustment.BoundaryType settings in the **GetOptionsResponse** message, then go to the next Parameter check.
8. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, IrCutFilterAutoAdjustment.BoundaryType = (Correct value from IrCutFilterAutoAdjustmentOptions, other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.



9. ONVIF Client will invoke **SetImagingSettingsRequest** message to set new imaging settings with the following parameters:
 - VideoSourceToken
 - IrCutFilterAutoAdjustment[0].BoundaryType=Correct value from IrCutFilterAutoAdjustmentOptions. If initial settings on the DUT do not contain all supported Boundary Types, then select value differs from values in initial settings.
 - IrCutFilterAutoAdjustment[0].BoundaryOffset=from +1.0 to -1.0. (other than current) if IrCutFilterAutoAdjustmentOptions.BoundaryOffset=true, otherwise skipped.
 - IrCutFilterAutoAdjustment[0].ResponseTime=Correct value from IrCutFilterAutoAdjustmentOptions
 - IrCutFilterAutoAdjustment[<other items>] skipped
 - Other imaging parameters from GetImagingSettingsResponse message
10. Verify the **SetImagingSettingsResponse** message from the DUT.
11. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
12. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that IrCutFilterAutoAdjustment.BoundaryType setting was applied.
13. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
14. Verify the **SetImagingSettingsResponse** message from the DUT.
15. If the DUT did not return IrCutFilterAutoAdjustment.BoundaryOffset is true in the **GetOptionsResponse** message, then go to the next Parameter check.
16. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, IrCutFilterAutoAdjustment.BoundaryOffset = from +1.0 to -1.0. (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
17. Verify the **SetImagingSettingsResponse** message from the DUT.
18. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
19. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that IrCutFilterAutoAdjustment.BoundaryOffset setting was applied.
20. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
21. Verify the **SetImagingSettingsResponse** message from the DUT.



22. If the DUT did not return `IrCutFilterAutoAdjustment.ResponseTimeRange` setting with `Max>Min` in the **GetOptionsResponse** message, then go to the next Parameter check.
23. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `IrCutFilterAutoAdjustment.ResponseTime` = (Correct value from `IrCutFilterAutoAdjustmentOptions`, other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
24. Verify the **SetImagingSettingsResponse** message from the DUT.
25. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
26. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that `IrCutFilterAutoAdjustment.ResponseTime` setting was applied.
27. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
28. Verify the **SetImagingSettingsResponse** message from the DUT.

29. If the DUT did not return at least two `ToneCompensation.Mode` setting in the **GetOptionsResponse** message, then go to the next Parameter check.
30. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `ToneCompensation.Mode` = enable/disable or automatic (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
31. Verify the **SetImagingSettingsResponse** message from the DUT.
32. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
33. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that setting was applied.
34. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
35. Verify the **SetImagingSettingsResponse** message from the DUT.

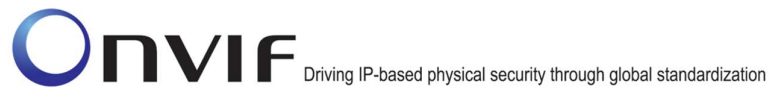
36. If the DUT did not return `ToneCompensation.Level` is true in the **GetOptionsResponse** message, then go to the next Parameter check.
37. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `ToneCompensation.Mode=on`, `ToneCompensation.Level` = value from 0.0 to +1.0 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
38. Verify the **SetImagingSettingsResponse** message from the DUT.



39. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
40. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that setting was applied.
41. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
42. Verify the **SetImagingSettingsResponse** message from the DUT.

43. If the DUT did not return at least two Defogging.Mode settings in the **GetOptionsResponse** message, then go to the next Parameter check.
44. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Defogging.Mode = On/off or Auto (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
45. Verify the **SetImagingSettingsResponse** message from the DUT.
46. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
47. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that setting was applied.
48. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
49. Verify the **SetImagingSettingsResponse** message from the DUT.

50. If the DUT did not return Defogging.Level is true in the **GetOptionsResponse** message, then go to the next Parameter check.
51. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, Defogging.Mode=On, Defogging.Level = value from 0.0 to +1.0 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
52. Verify the **SetImagingSettingsResponse** message from the DUT.
53. ONVIF Client will invoke **GetImagingSettingsRequest** message (VideoSourceToken) to retrieve current imaging settings.
54. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that setting was applied.
55. ONVIF Client will invoke **SetImagingSettingsRequest** message (VideoSourceToken, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
56. Verify the **SetImagingSettingsResponse** message from the DUT.



57. If the DUT did not return `NoiseReduction.Level` is true in the **GetOptionsResponse** message, then go to the next Parameter check.
58. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, `NoiseReduction.Level` = value from 0.0 to +1.0 (other than current), other imaging parameters from **GetImagingSettingsResponse** message) to set new imaging settings.
59. Verify the **SetImagingSettingsResponse** message from the DUT.
60. ONVIF Client will invoke **GetImagingSettingsRequest** message (`VideoSourceToken`) to retrieve current imaging settings.
61. Verify the **GetImagingSettingsResponse** message from the DUT. Verify that setting was applied.
62. ONVIF Client will invoke **SetImagingSettingsRequest** message (`VideoSourceToken`, all imaging parameters from **GetImagingSettingsResponse** message at step 6) to return imaging settings to initial state.
63. Verify the **SetImagingSettingsResponse** message from the DUT.



4.2 Focus Move

4.2.1 IMAGING COMMAND GETMOVEOPTIONS

Test Label: Imaging DUT GetMoveOptions Command Validation.

Test Case ID: IMAGING-2-1-1

ONVIF Core Specification Coverage: Get move options

Command under Test: GetMoveOptions

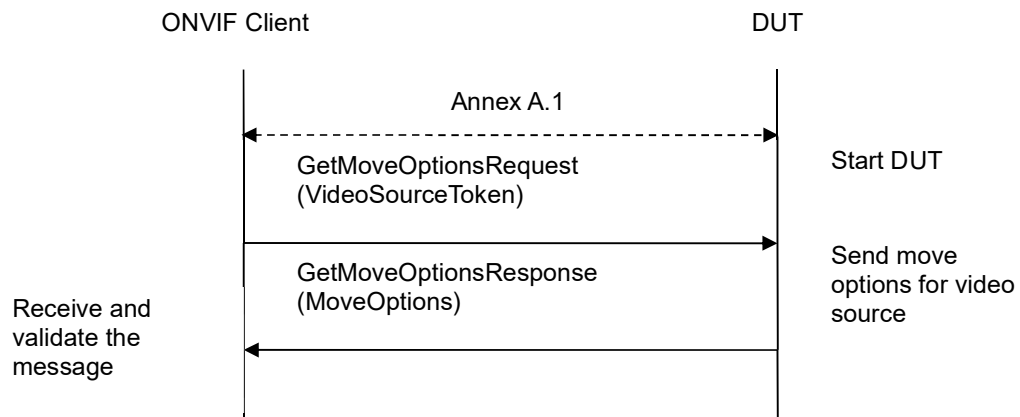
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetMoveOptions command.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. ONVIF Client will repeat steps 3-4 for other video sources supported by the DUT.

Test Results:

PASS –

DUT passed all assertions.

FAIL –



The DUT did not send `GetMoveOptionsResponse` message.

The DUT did not send valid `GetMoveOptionsResponse` message

The DUT returned `GetMoveOptionsResponse` message with Min value greater than Max for at least one setting defined by Min and Max boundary:

- `Absolute.Position`
- `Absolute.Speed`
- `Relative.Distance`
- `Relative.Speed`
- `Continuous.Speed`

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.2 IMAGING COMMAND ABSOLUTE MOVE

Test Label: Imaging DUT Move Command Validation for Absolute Move.

Test Case ID: IMAGING-2-1-3

ONVIF Core Specification Coverage: Move

Command under Test: Move

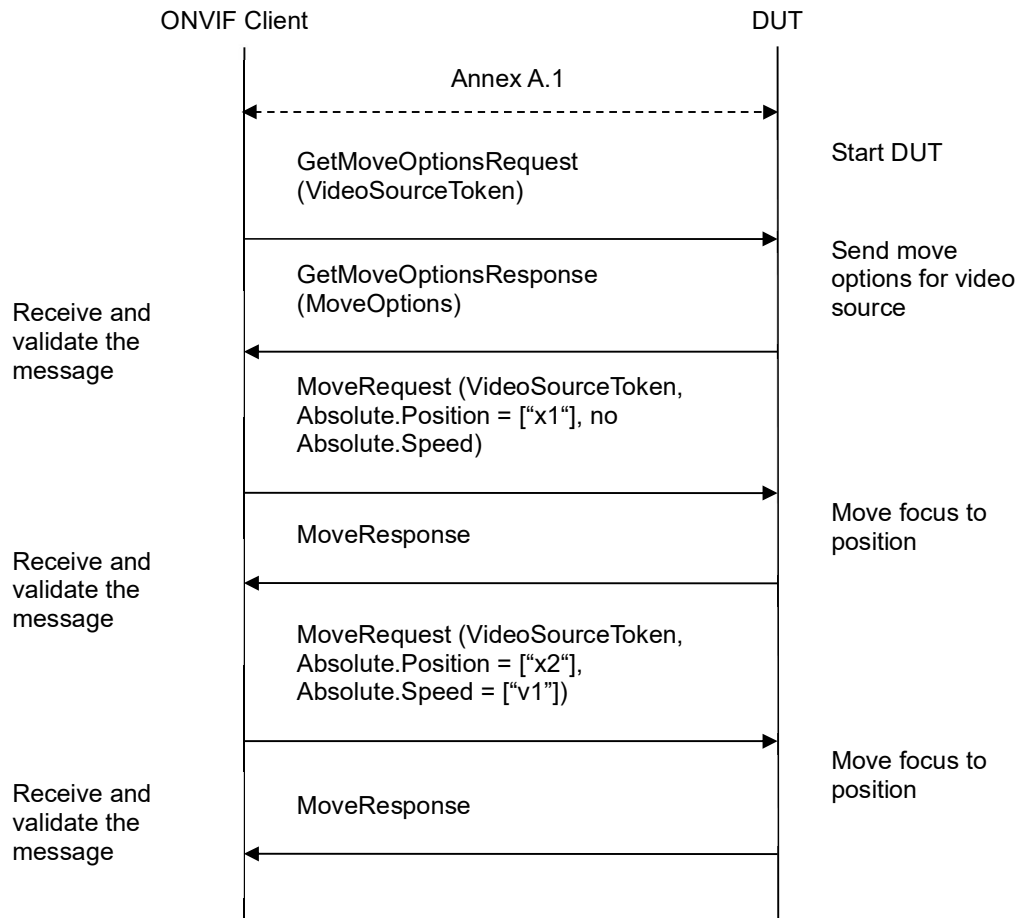
WSDL Reference: `imaging.wsdl`

Test Propose: To verify the behavior of Move command in case of Absolute Move.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke `GetMoveOptionsRequest` message (`VideoSourceToken`) to retrieve current move options for Video Source.
4. Verify the `GetMoveOptionsResponse` message from the DUT.
5. If Absolute Move is not supported by Video Source (there is no Absolute tag in `GetMoveOptionsResponse` message), go to step 10.
6. ONVIF Client will invoke `MoveRequest` message (`VideoSourceToken`, `Absolute.Position = ["x1"]`, no `Absolute.Speed`) to move focus to position.
7. Verify that the DUT has returned a valid `MoveResponse` message.
8. If `Absolute.Speed` is supported by Video Source (there is tag `Absolute.Speed` in `GetMoveOptionsResponse` message), ONVIF Client will invoke `MoveRequest` message (`VideoSourceToken`, `Absolute.Position = ["x2"]`, `Absolute.Speed = ["v1"]`) to move focus to position with the specified speed, else go to the step 10.



9. Verify that the DUT has returned a valid MoveResponse message.

10. ONVIF Client will repeat steps 3-9 for other video sources supported by the DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send MoveResponse message.

The DUT did not send a valid MoveResponse message.

4.2.3 IMAGING COMMAND ABSOLUTE MOVE – INVALID SETTINGS

Test Label: Imaging DUT Move Command Validation for Absolute Move (Invalid Settings).

Test Case ID: IMAGING-2-1-4

ONVIF Core Specification Coverage: Move

Command under Test: Move

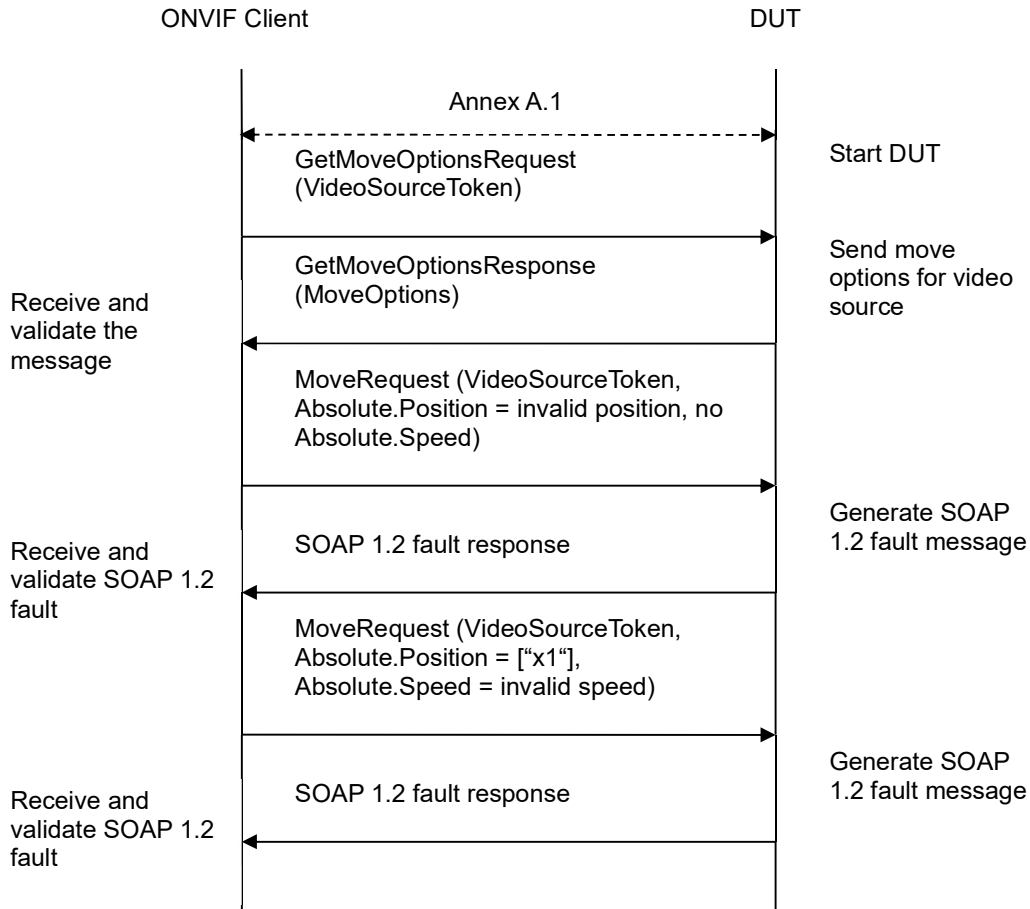
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of Absolute Move and invalid settings.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Absolute Move is not supported by Video Source (there is no Absolute tag in GetMoveOptionsResponse message), go to step 10.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Absolute.Position = invalid position, no Absolute.Speed).
7. The DUT will generate a SOAP 1.2 fault message.
8. If Absolute.Speed is supported by Video Source (there is tag Absolute.Speed in GetMoveOptionsResponse message) ONVIF Client will invoke MoveRequest message (VideoSourceToken, Absolute.Position = [\"x1\", Absolute.Speed = invalid speed), else go to the step 10.



9. The DUT will generate a SOAP 1.2 fault message.

10. ONVIF Client will repeat steps 5-9 for other video sources supported by DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send SOAP 1.2 fault message for MoveRequest.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.4 IMAGING COMMAND RELATIVE MOVE

Test Label: Imaging DUT Move Command Validation for Relative Move.

Test Case ID: IMAGING-2-1-5

ONVIF Core Specification Coverage: Move

Command under Test: Move

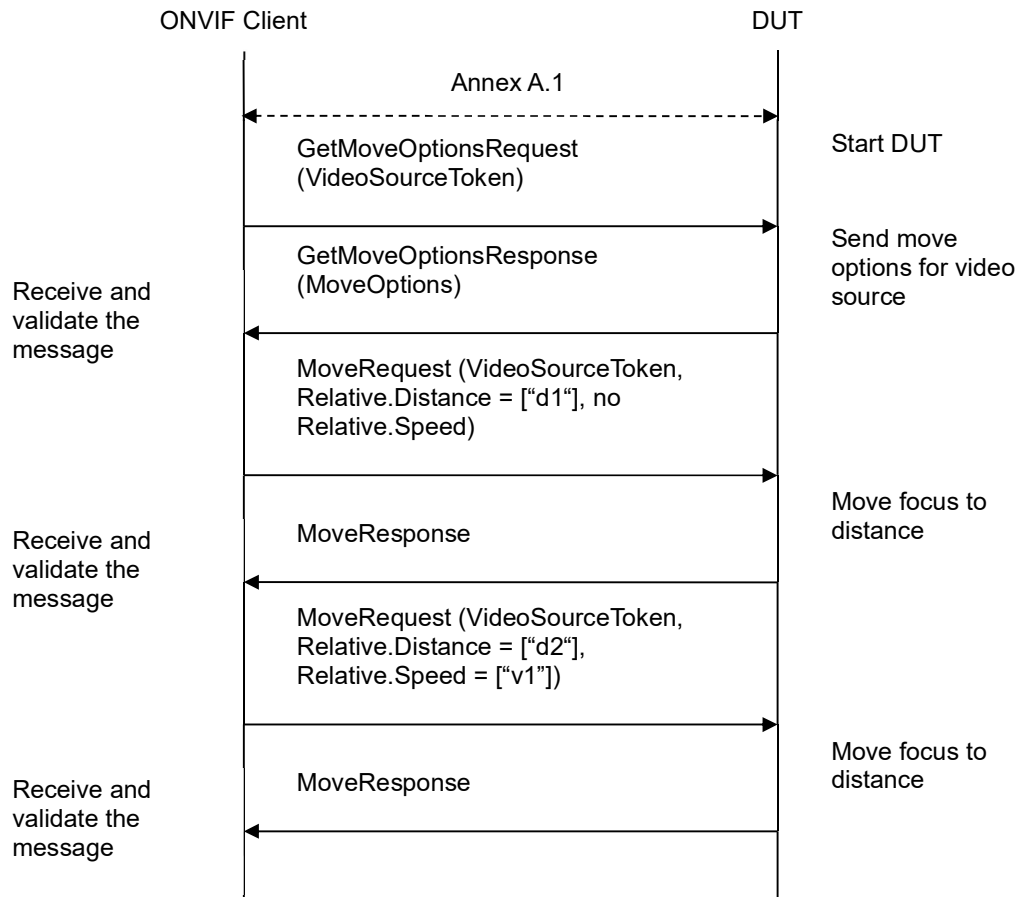
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of Relative Move.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Relative Move is not supported by Video Source (there is no Relative tag in GetMoveOptionsResponse message) go to step 10.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Relative.Distance = ["d1"], no Relative.Speed) to move focus to a distance.
7. Verify that the DUT has returned a valid MoveResponse message.
8. If Relative.Speed is supported by Video Source (there is tag Relative.Speed in GetMoveOptionsResponse message) ONVIF Client will invoke MoveRequest message (VideoSourceToken, Relative.Distance = ["d2"], Relative.Speed = ["v1"]) to move focus to a distance with specified speed, else go to the step 10.



9. Verify that the DUT has returned a valid MoveResponse message.

10. ONVIF Client will repeat steps 3-9 for other video sources supported by the DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send MoveResponse message.

The DUT did not send a valid MoveResponse message.

4.2.5 IMAGING COMMAND RELATIVE MOVE – INVALID SETTINGS

Test Label: Imaging DUT Move Command Validation for Relative Move (Invalid Settings).

Test Case ID: IMAGING-2-1-6

ONVIF Core Specification Coverage: Move

Command under Test: Move

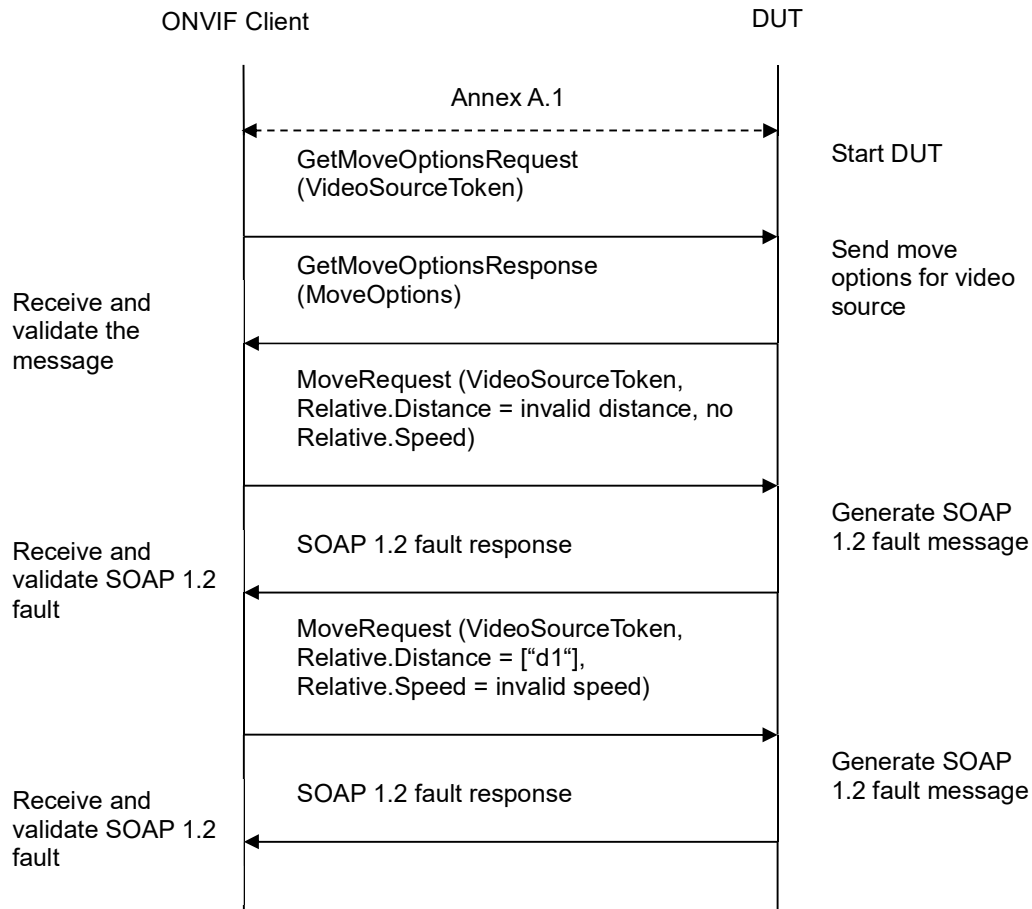
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of Relative Move and invalid settings.

Pre-Requisite: None

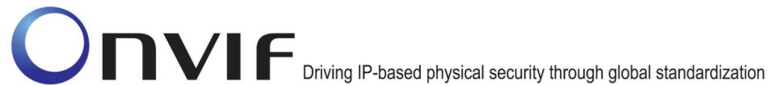
Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Relative Move is not supported by Video Source (there is no Relative tag in GetMoveOptionsResponse message) go to step 10.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Relative.Distance = invalid distance, no Relative.Speed).
7. The DUT will generate a SOAP 1.2 fault message.
8. If Relative.Speed is supported by Video Source (there is tag Relative.Speed in GetMoveOptionsResponse message) ONVIF Client will invoke MoveRequest message (VideoSourceToken, Relative.Distance = [\"d1\", Relative.Speed = invalid speed), else go to the step 10.



9. The DUT will generate a SOAP 1.2 fault message.

10. ONVIF Client will repeat steps 3-9 for other video sources supported by DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send SOAP 1.2 fault message for MoveRequest.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.6 IMAGING COMMAND CONTINUOUS MOVE

Test Label: Imaging DUT Move Command Validation for Continuous Move.

Test Case ID: IMAGING-2-1-7

ONVIF Core Specification Coverage: Move

Command under Test: Move, Stop

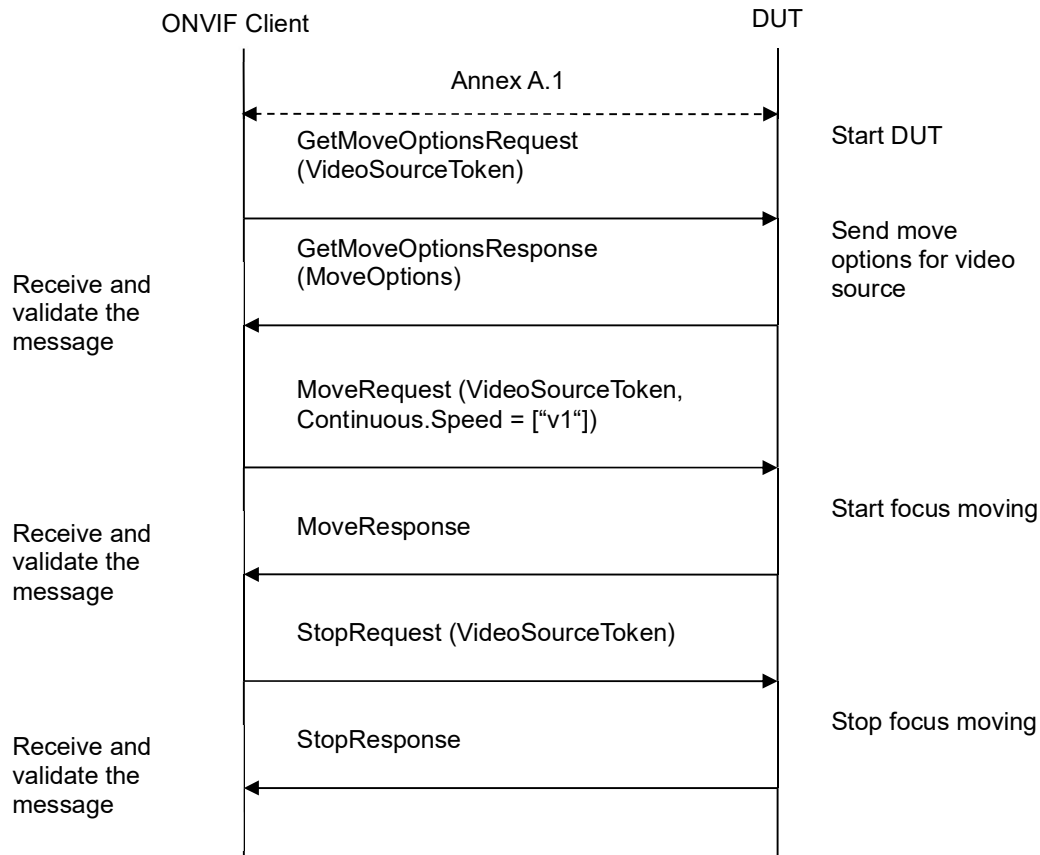
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of Continuous Move.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Continuous Move is not supported by Video Source (there is no Continuous tag in GetMoveOptionsResponse message), go to step 10.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Continuous.Speed = ["v1"]) to start focus moving.
7. Verify that the DUT has returned a valid MoveResponse message.
8. ONVIF Client will invoke StopRequest message (VideoSourceToken) to stop focus moving.
9. Verify that the DUT has returned a valid StopResponse message.
10. ONVIF Client will repeat steps 3-10 for other video sources supported by the DUT.



Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send MoveResponse message.

The DUT did not send a valid MoveResponse message.

The DUT did not send StopResponse message.

The DUT did not send a valid StopResponse message.

Note: If Stop command is not supported by Video Source, ONVIF Client will use timeout to wait until the DUT stops moving.

4.2.7 IMAGING COMMAND CONTINUOUS MOVE – INVALID SETTINGS

Test Label: Imaging DUT Move Command Validation for Continuous Move (Invalid Settings).

Test Case ID: IMAGING-2-1-8

ONVIF Core Specification Coverage: Move

Command under Test: Move

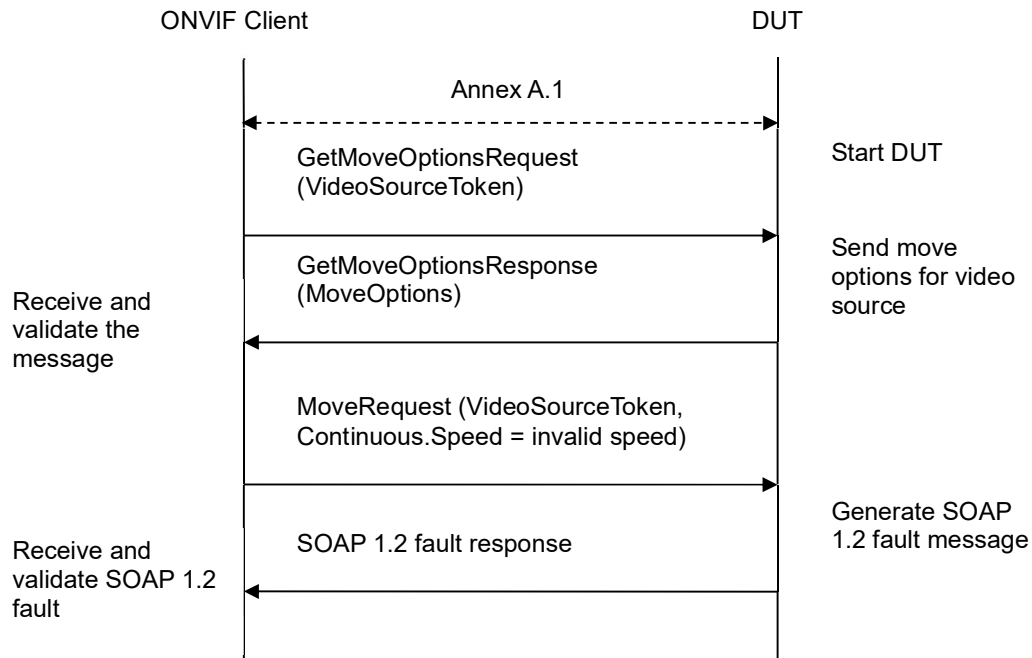
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of Continuous Move and invalid settings.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Continuous Move is not supported by Video Source (there is no Continuous tag in GetMoveOptionsResponse message), go to step 8.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Continuous.Speed = invalid speed).
7. The DUT will generate a SOAP 1.2 fault message.
8. ONVIF Client will repeat steps 3-7 for other video sources supported by DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send SOAP 1.2 fault message for MoveRequest.



The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.8 IMAGING COMMAND MOVE – UNSUPPORTED MOVE

Test Label: Imaging DUT Move Command Validation (Unsupported Move).

Test Case ID: IMAGING-2-1-10

ONVIF Core Specification Coverage: Move

Command under Test: Move

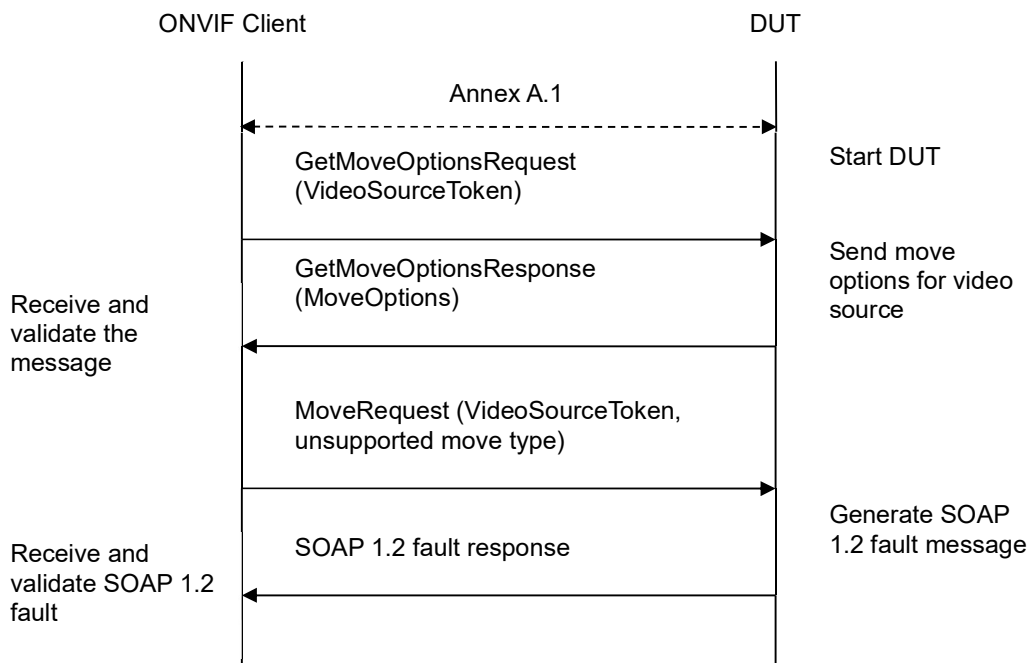
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Move command in case of unsupported move request.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.



3. ONVIF Client will invoke GetMoveOptionsRequest message (VideoSourceToken) to retrieve current move options for Video Source.
4. Verify the GetMoveOptionsResponse message from the DUT.
5. If Absolute Move is supported by Video Source (there is Absolute tag in GetMoveOptionsResponse message), go to step 8.
6. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Absolute.Position = any value, Absolute.Speed = any value).
7. The DUT will generate a SOAP 1.2 fault message.
8. If Relative Move is supported by Video Source (there is Relative tag in GetMoveOptionsResponse message), go to step 11.
9. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Relative.Distance = any value, Relative.Speed = any value).
10. The DUT will generate a SOAP 1.2 fault message.
11. If Continuous Move is supported by Video Source (there is Continuous tag in GetMoveOptionsResponse message), go to step 14.
12. ONVIF Client will invoke MoveRequest message (VideoSourceToken, Continuous.Speed = any value).
13. The DUT will generate a SOAP 1.2 fault message.
14. ONVIF Client will repeat steps 3-13 for other video sources supported by DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetMoveOptionsResponse message.

The DUT did not send a valid GetMoveOptionsResponse message.

The DUT did not send SOAP 1.2 fault message for MoveRequest.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.9 IMAGING COMMAND GETSTATUS

Test Label: Imaging DUT GetStatus Command Validation.

Test Case ID: IMAGING-2-1-11

ONVIF Core Specification Coverage: Get imaging status



Command under Test: GetStatus

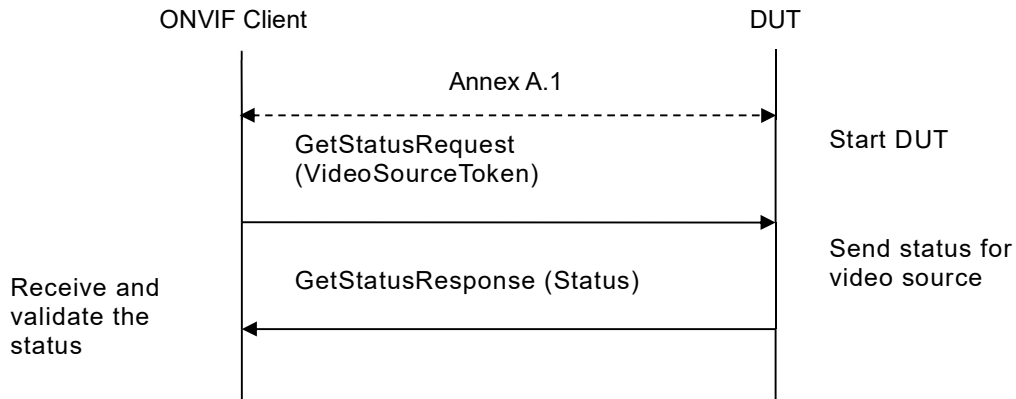
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetStatus command.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetStatusRequest message (VideoSourceToken) to retrieve current status for Video Source.
4. If the DUT generates a SOAP 1.2 fault message (ActionNotSupported/NoImagingForSource (in case Imaging is not supported for Video Source) or ActionNotSupported (in case the GetStatus command is not implemented)), verify it.
5. If the DUT returns status for Video Source in the GetStatusResponse message, verify the GetStatusResponse message from the DUT.
6. ONVIF Client will repeat steps 3-5 for other video sources supported by the DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send GetStatusResponse message or send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

The DUT did not send GetStatusResponse message or send SOAP 1.2 fault message.



The DUT sent an invalid GetStatusResponse message.

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.10 IMAGING COMMAND STOP

Test Label: Imaging DUT Stop Command Validation.

Test Case ID: IMAGING-2-1-13

ONVIF Core Specification Coverage: Stop

Command under Test: Stop

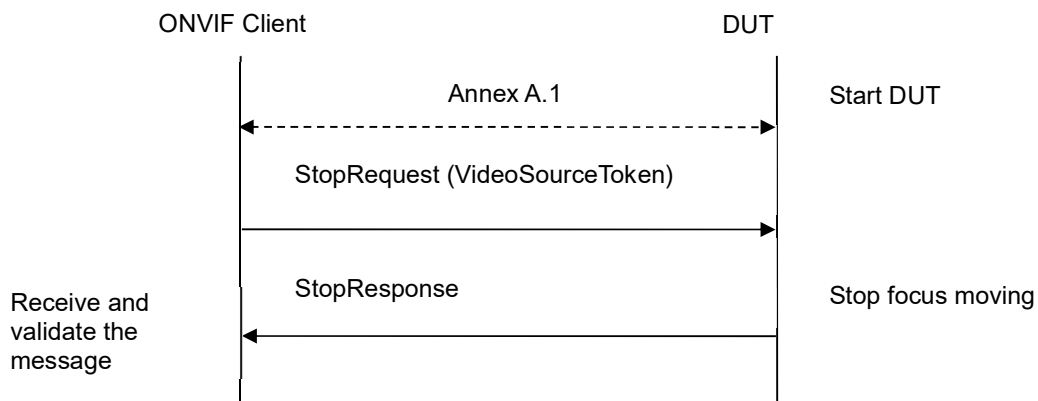
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Stop command.

Pre-Requisite: None

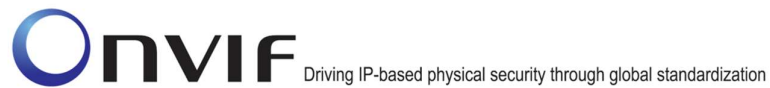
Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke StopRequest message (VideoSourceToken) to stop movement for Video Source.
4. If the DUT generates a SOAP 1.2 fault message (ActionNotSupported/NoImagingForSource (in case Imaging is not supported for Video Source) or ActionNotSupported (in case the Stop command is not implemented)), verify it.



5. If the DUT returns the StopResponse message, verify the StopResponse message from the DUT.
6. ONVIF Client will repeat steps 3-5 for other video sources supported by the DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send StopResponse message or send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

The DUT sent an invalid StopResponse message.

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

4.2.11 IMAGING COMMAND GETMOVEOPTIONS – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT GetMoveOptions Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-2-1-15

ONVIF Core Specification Coverage: Get move options

Command under Test: GetMoveOptions

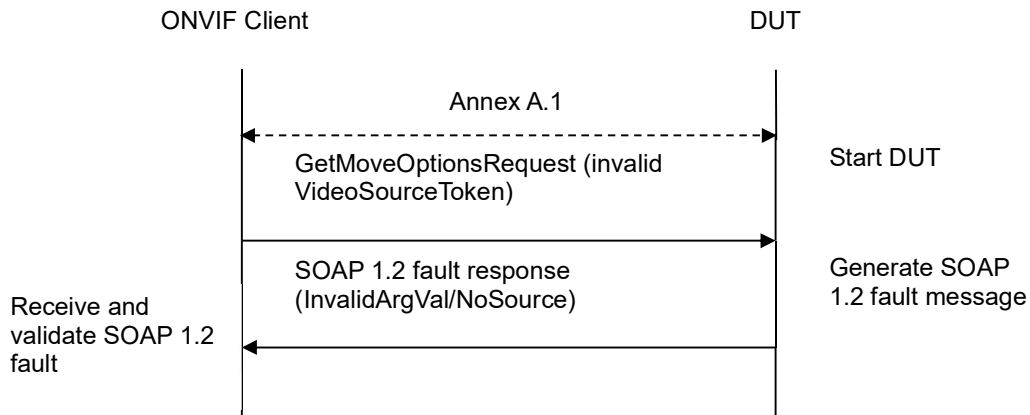
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetMoveOptions command in case of invalid VideoSourceToken.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetMoveOptionsRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.

4.2.12 IMAGING COMMAND MOVE – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT Move Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-2-1-16

ONVIF Core Specification Coverage: Move

Command under Test: Move

WSDL Reference: imaging.wsdl

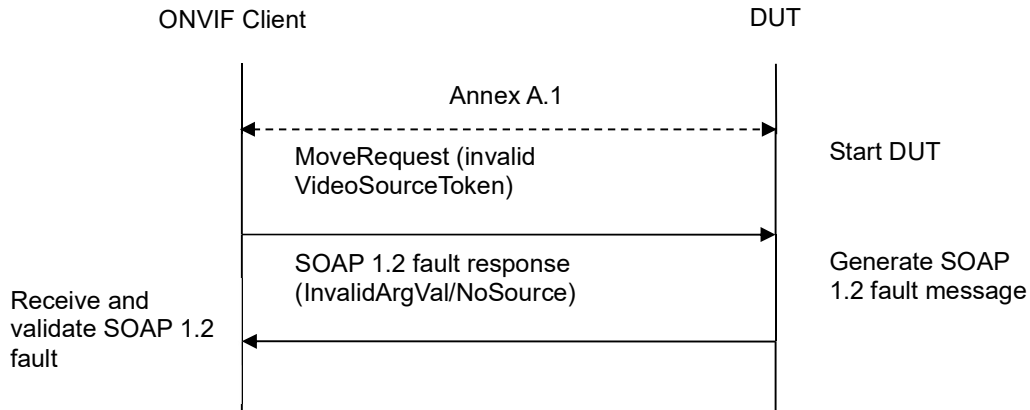
Test Propose: To verify the behavior of Move command in case of invalid VideoSourceToken.



Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke MoveRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.

4.2.13 IMAGING COMMAND GETSTATUS – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT GetStatus Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-2-1-17

ONVIF Core Specification Coverage: Get imaging status



Command under Test: GetStatus

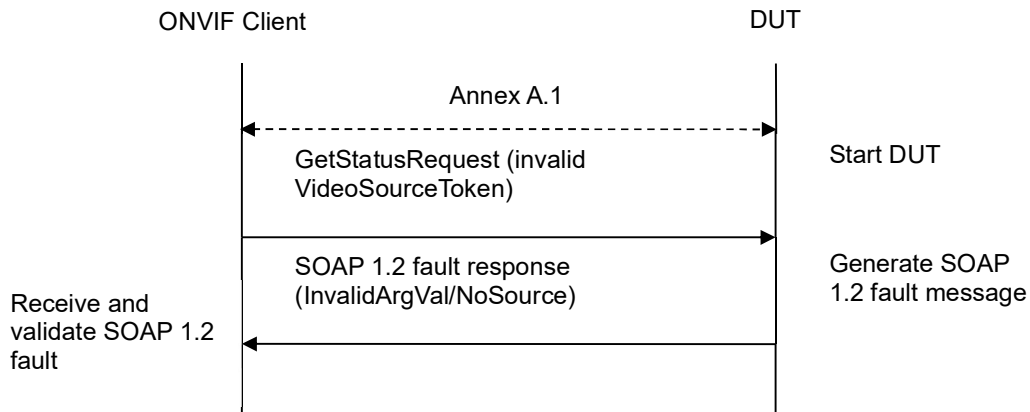
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of GetStatus command in case of invalid **VideoSourceToken**.

Pre-Requisite: None

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke GetStatusRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource or ActionNotSupported (for the case when GetStatus command is not implemented)).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).

Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.



4.2.14 IMAGING COMMAND STOP – INVALID VIDEOSOURCETOKEN

Test Label: Imaging DUT Stop Command Validation (Invalid VideoSourceToken).

Test Case ID: IMAGING-2-1-18

ONVIF Core Specification Coverage: Stop

Command under Test: Stop

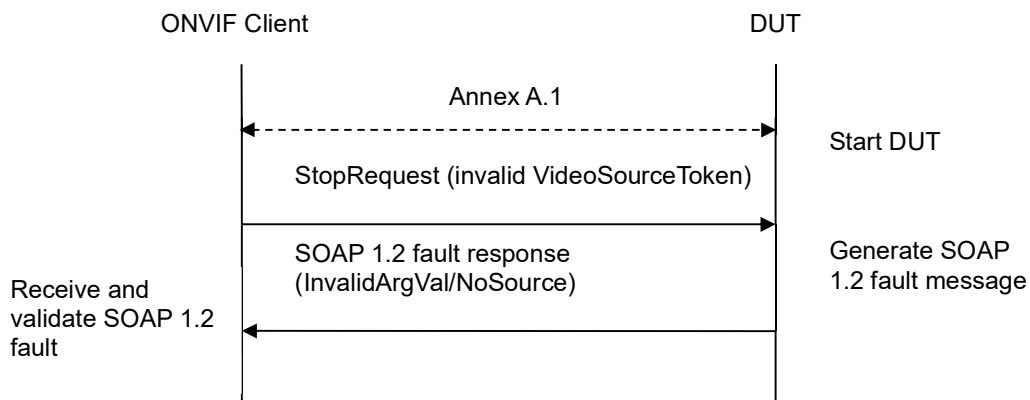
WSDL Reference: imaging.wsdl

Test Propose: To verify the behavior of Stop command in case of invalid VideoSourceToken.

Test Configuration: ONVIF Client and DUT

Pre-Requisite: None

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start a DUT.
3. ONVIF Client will invoke StopRequest message with an invalid VideoSourceToken.
4. The DUT will generate a SOAP 1.2 fault message (InvalidArgVal/NoSource or ActionNotSupported (in case the Stop command is not implemented)).

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send SOAP 1.2 fault message.

The DUT sent an incorrect SOAP 1.2 fault message (fault code, namespace, etc.).



Note: See Annex in [ONVIF Base Test] for invalid SOAP 1.2 fault message definition.

Note: Other faults than specified in the test are acceptable but the specified are preferable.

Note: See Annex A.2 for Name and Token Parameters Length limitations.



4.3 Capabilities

4.3.1 IMAGING SERVICE CAPABILITIES

Test Label: Imaging Service Capabilities Verification.

Test Case ID: IMAGING-3-1-1

ONVIF Core Specification Coverage: Capability exchange

Command under test: GetServiceCapabilities (for Imaging Service)

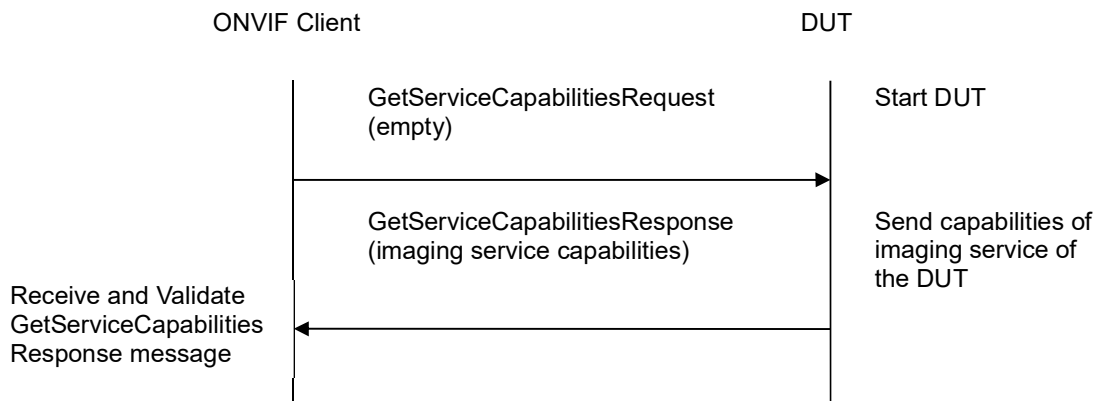
WSDL Reference: imaging.wsdl

Test Purpose: To verify Imaging Service Capabilities of the DUT.

Pre-Requisite: Imaging Service was received from the DUT.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client will invoke GetServiceCapabilitiesRequest message to retrieve Imaging service capabilities of the DUT.
4. Verify the GetServiceCapabilitiesResponse message from the DUT.

Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send valid GetServiceCapabilitiesResponse.



4.3.2 GET SERVICES AND GET IMAGING SERVICE CAPABILITIES CONSISTENCY

Test Label: Get Services and Imaging Service Capabilities Consistency Verification.

Test Case ID: IMAGING-3-1-2

ONVIF Core Specification Coverage: Capability exchange

Command under test: GetServices, GetServiceCapabilities (for Imaging Service)

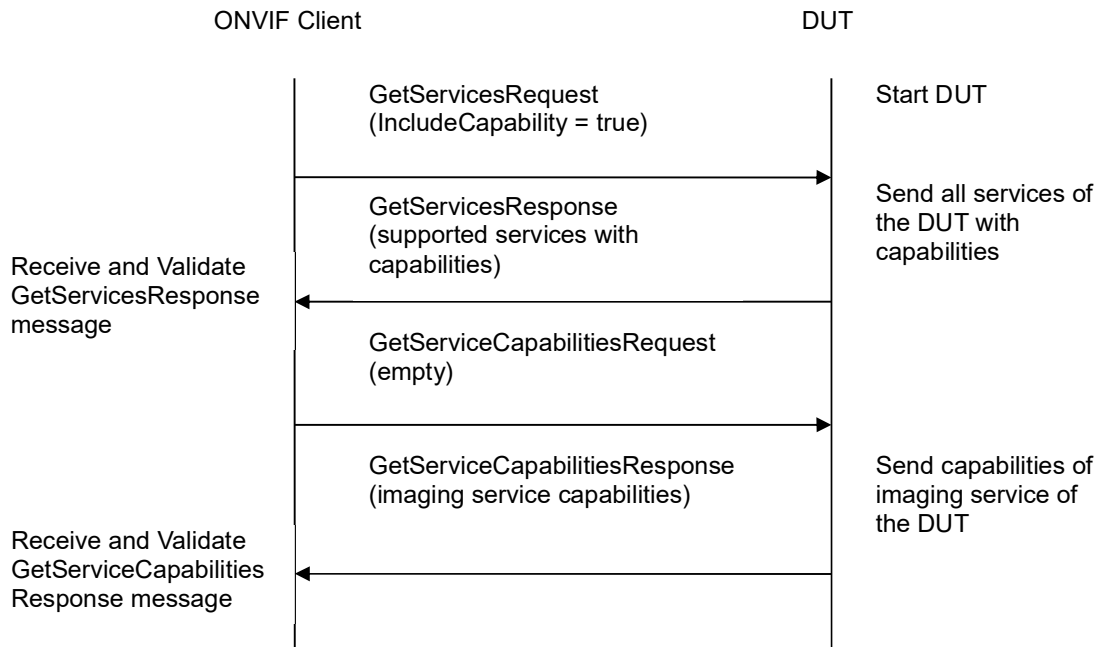
WSDL Reference: devicemgmt.wsdl, imaging.wsdl

Test Purpose: To verify Get Services and Imaging Service Capabilities consistency.

Pre-Requisite: None.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client will invoke GetServicesRequest message (IncludeCapability = true) to retrieve all services of the DUT with service capabilities.
4. Verify the GetServicesResponse message from the DUT.
5. ONVIF Client will invoke GetServiceCapabilitiesRequest message to retrieve Imaging service capabilities of the DUT.
6. Verify the GetServiceCapabilitiesResponse message from the DUT.



Test Result:

PASS –

DUT passed all assertions.

FAIL –

The DUT did not send a valid GetServicesResponse message.

The DUT did not send a valid GetServiceCapabilitiesResponse message.

The DUT sent different Capabilities in GetServicesResponse message and in GetServiceCapabilitiesResponse message.

Note: Service will be defined as Imaging service if it will have Namespace element that is equal to “<http://www.onvif.org/ver20/imaging/wsd>”.



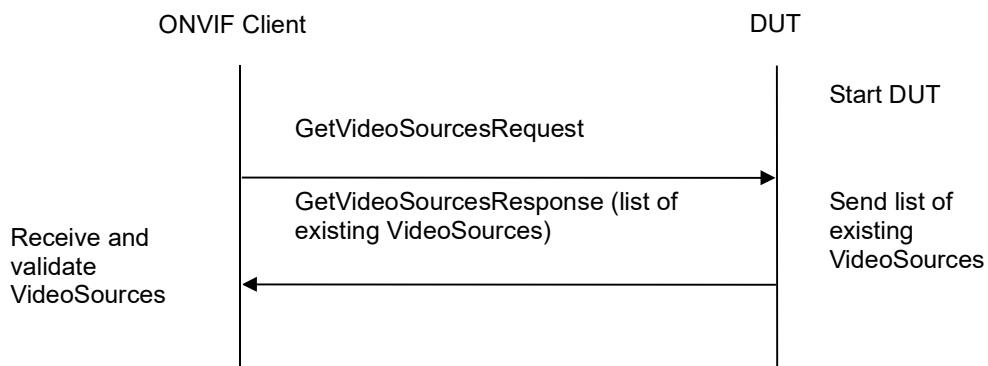
Annex A

This section describes the meaning of the following definitions. These definitions are used in the test case description.

A.1 Get Video Sources

For the execution of Imaging test cases, ONVIF Client has to obtain the list of Video Sources for retrieving available Video Source as follows.

1. For obtaining the list of Video Source ONVIF Client invokes GetVideoSourcesRequest for DeviceIO service, otherwise for Media service.
2. DUT sends GetVideoSourcesResponse with the list of Video Source. Then ONVIF Client verifies the response and uses VideoSourceToken of each VideoSources for the following test case.



A.2 Name and Token Parameters

There are following limitations on maximum length of Name and Token parameters that shall be used during tests by ONVIF Device Test Tool to prevent faults from DUT:

1. Name shall be less than or equal to 64 characters (only readable characters accepted).
2. Token shall be less than or equal to 64 characters (only readable characters accepted).

UTF-8 character set shall be used for Name and Token.

Note: these limitations will not be used, if ONVIF Device Test Tool reuses values that were received from the DUT.