

# **ONVIF<sup>®</sup>**

# **Provisioning Device Test Specification**

Version 18.06

June 2018

© 2018 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

<b>Vers.</b>	<b>Date</b>	<b>Description</b>
17.12	Aug 1, 2017	Initial version
18.06	Jun 21, 2018	Reformatting document using new template

**Table of Contents**

- 1 Introduction ..... 5**
  - 1.1 Scope ..... 5
    - 1.1.1 Provisioning Move ..... 6
    - 1.1.2 Capabilities ..... 6
- 2 Normative references ..... 7**
- 3 Terms and Definitions ..... 9**
  - 3.1 Conventions ..... 9
  - 3.2 Definitions ..... 9
- 4 Test Overview ..... 10**
  - 4.1 Test Setup ..... 10
    - 4.1.1 Network Configuration for DUT ..... 10
  - 4.2 Prerequisites ..... 11
  - 4.3 Test Policy ..... 11
    - 4.3.1 Provisioning Move ..... 11
    - 4.3.2 Capabilities ..... 12
- 5 Provisioning Test Cases ..... 13**
  - 5.1 Provisioning Move ..... 13
    - 5.1.1 PROVISIONING PAN MOVE AND STOP OPERATION ..... 13
    - 5.1.2 PROVISIONING TILT MOVE AND STOP OPERATION ..... 15
    - 5.1.3 PROVISIONING ZOOM MOVE AND STOP OPERATION ..... 17
    - 5.1.4 PROVISIONING ROLL MOVE AND STOP OPERATION ..... 19
    - 5.1.5 PROVISIONING FOCUS MOVE AND STOP OPERATION ..... 22
  - 5.2 Capabilities ..... 25
    - 5.2.1 GET SERVICES AND GET PROVISIONING SERVICE CAPABILITIES  
CONSISTENCY ..... 25
- A Helper Procedures and Additional Notes ..... 28**
  - A.1 Get Provisioning Service Capabilities ..... 28

# 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 need to verify the [ONVIF Provisioning Service Specs] and [ONVIF Conformance] requirements. It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.

This ONVIF Provisioning Test Specification acts as a supplementary document to the [ONVIF Provisioning Service Specs], illustrating test cases need to be executed and passed. And 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 Provisioning 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 test cases to test individual requirements of ONVIF devices according to ONVIF Provisioning Service which is defined in [ONVIF Provisioning Service 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 Service Interoperability Basic Profile version 2.0 (WS-I BP 2.0).
3. Network protocol implementation Conformance test for HTTP, HTTPS, RTP protocol.
4. Wi-Fi Conformance test

The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Provisioning Service Specs]; instead it would cover subset of it. The scope of this specification is to derive all the normative requirements of [ONVIF Provisioning Service Specs] which are related to ONVIF Provisioning Service and some of the optional requirements.

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

### 1.1.1 Provisioning Move

The Provisioning Move section covers the test cases needed for adjustments that are not intended for general day-to-day use.

The scope of this specification section is to cover the following functions:

- Moving device on the focus axis with FocusMove command
- Moving device on the pan axis with PanMove command
- Moving device on the roll axis with RollMove command
- Moving device on the tilt axis with TiltMove command
- Moving device on the zoom axis with ZoomMove command
- Stopping movement on all provisioning methods with Stop command
- Returning the lifetime move counts with GetUsage command

### 1.1.2 Capabilities

The Capabilities section covers the test cases needed for getting capabilities from an ONVIF device.

The scope of this specification section is to cover the following functions:

- Getting Provisioning service address with GetServices command via Device service
- Getting capabilities with GetServiceCapabilities command
- Getting capabilities with GetServices command via Device service

## 2 Normative references

- [ONVIF Conformance] ONVIF Conformance Process Specification:  
<https://www.onvif.org/profiles/conformance/>
- [ONVIF Profile Policy] ONVIF Profile Policy:  
<https://www.onvif.org/profiles/>
- [ONVIF Network Interface Specs] ONVIF Network Interface Specification documents:  
<https://www.onvif.org/profiles/specifications/>
- [ONVIF Core Specs] ONVIF Core Specifications:  
<https://www.onvif.org/profiles/specifications/>
- [ONVIF Provisioning Service] Provisioning Service Specifications:  
<https://www.onvif.org/profiles/specifications/>
- [ONVIF Base Test] ONVIF Base Device Test Specification:  
<https://www.onvif.org/profiles/conformance/device-test/>
- [ISO/IEC Directives, Part 2] ISO/IEC Directives, Part 2, Annex H:  
<http://www.iso.org/directives>
- [ISO 16484-5] ISO 16484-5:2014-09 Annex P:  
<https://www.iso.org/obp/ui/#iso:std:63753:en>
- [SOAP 1.2, Part 1] W3C SOAP 1.2, Part 1, Messaging Framework:  
<http://www.w3.org/TR/soap12-part1/>
- [XML-Schema, Part 1] W3C XML Schema Part 1: Structures Second Edition:  
<http://www.w3.org/TR/xmlschema-1/>
- [XML-Schema, Part 2] W3C XML Schema Part 2: Datatypes Second Edition:  
<http://www.w3.org/TR/xmlschema-2/>
- [WS-Security] "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard, February 2006.:

<http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>



## 3 Terms and Definitions

### 3.1 Conventions

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

### 3.2 Definitions

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

<b>Provisioning</b>	Doing something in advance to prepare for something else.
<b>Video Source</b>	Entity defined by [ONVIF Device I/O Specification]
<b>Video Source Token</b>	Token referencing a Device I/O Video Source

## 4 Test Overview

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

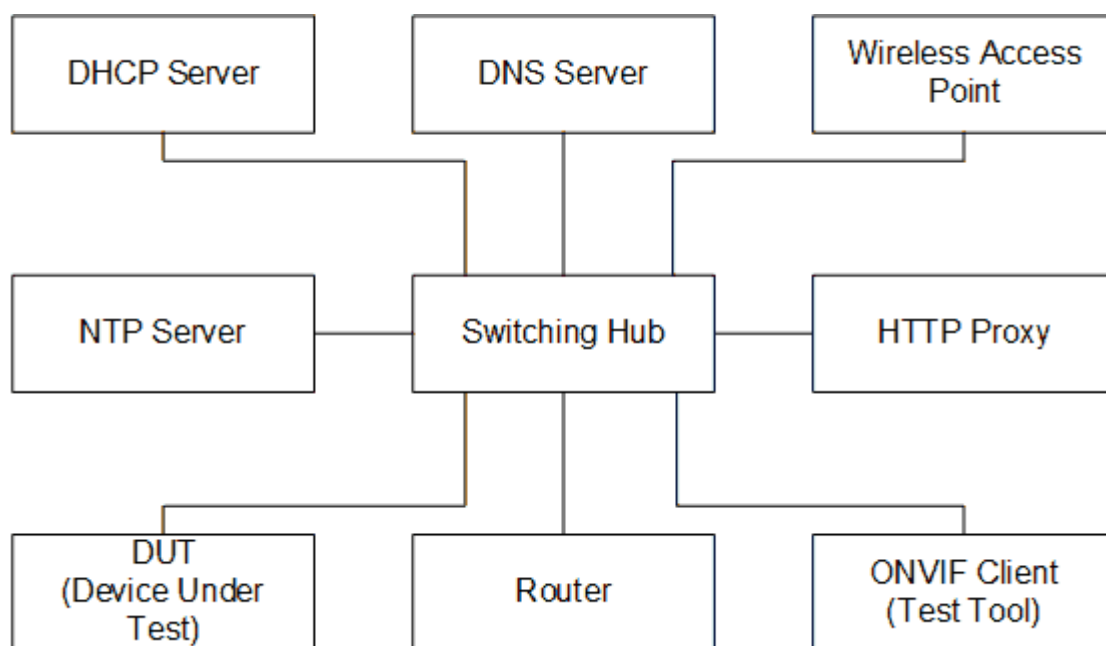
### 4.1 Test Setup

#### 4.1.1 Network Configuration for DUT

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 4.1).

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 4.1. 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 behavior of the DUT. It handles both expected and unexpected behavior.

**HTTP Proxy:** provides facilitation in case of RTP and RTSP tunneling 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. When running multiple test instances in parallel on the same network, the Switching Hub should be configured to use filtering in order to avoid multicast traffic being flooded to all ports, because this may affect test stability.

**Router:** provides router advertisements for IPv6 configuration.

## 4.2 Prerequisites

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

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

## 4.3 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.

### 4.3.1 Provisioning Move

The test policies specific to the test case execution of all functional blocks:

- DUT shall give the Provisioning Service entry point by GetServices command, if DUT supports this service. Otherwise, these test cases will be skipped.
- DUT shall support the following commands:
  - GetUsage

- Stop
- FocusMove (if supported)
- PanMove (if supported)
- RollMove (if supported)
- TiltMove (if supported)
- ZoomMove (if supported)
- After each provisioning operation the corresponding usage number shall be increased. The current value is determined by executing the GetUsage command.

Please refer to [Section 5.1](#) for Provisioning Move Test Cases.

## 4.3.2 Capabilities

The test policies specific to the test case execution of Capabilities functional block:

- DUT shall give the Provisioning Service entry point by GetServices command, if DUT supports this service. Otherwise, these test cases will be skipped.
- DUT shall support the following commands:
  - GetServices
  - GetServiceCapabilities
- The following tests are performed
  - Getting capabilities with GetServiceCapabilities command
  - Getting capabilities with GetServices command

Please refer to [Section 5.2](#) for Provisioning Move Test Cases.

## 5 Provisioning Test Cases

### 5.1 Provisioning Move

#### 5.1.1 PROVISIONING PAN MOVE AND STOP OPERATION

**Test Case ID:** PROVISIONING-1-1-1

**Specification Coverage:** Pan Move (Provisioning Service Specification), Stop (Provisioning Service Specification)

**Feature Under Test:** PanMove, Stop

**WSDL Reference:** provisioning.wsdl

**Test Purpose:** To verify Pan movement and stopping using the DUT Provisioning PanMove operation.

**Pre-Requisite:** Provisioning Service is received from the DUT.

**Test Configuration:** ONVIF Client and DUT

**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client retrieves Provisioning Service capabilities by following the procedure mentioned in [Annex A.1](#) with the following input and output parameters
  - out *cap* - Provisioning Service capabilities
4. Set *timeout1* := "P2D"
5. For each Source *cap.Source* in *cap* repeat the following steps:
  - 5.1. If *cap.Source.MaximumPanMoves* is present:
    - 5.1.1. Set *videoSourceToken1* := *cap.Source.@VideoSourceToken*
    - 5.1.2. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.3. The DUT responds with **GetUsageResponse** with parameters
      - *Usage* := *usage1*

- 5.1.4. If there is no *usage1.Pan* item, FAIL the test and skip other steps.
- 5.1.5. ONVIF Client invokes **PanMove** request with parameters
  - VideoSource := *videoSourceToken1*
  - Direction := "Left"
  - Timeout skipped
- 5.1.6. The DUT responds with **PanMoveResponse** message.
- 5.1.7. Set *defaultTimeout* := *cap.DefaultTimeout*
- 5.1.8. Wait until *defaultTimeout* timeout expires.
- 5.1.9. ONVIF Client invokes **GetUsage** request with parameters
  - VideoSource := *videoSourceToken1*
- 5.1.10. The DUT responds with **GetUsageResponse** with parameters
  - Usage := *usage2*
- 5.1.11. If *usage2.Pan* <= *usage1.Pan* item, FAIL the test and skip other steps.
- 5.1.12. ONVIF Client invokes **PanMove** request with parameters
  - VideoSource := *videoSourceToken1*
  - Direction := "Right"
  - Timeout := *timeout1*
- 5.1.13. The DUT responds with **PanMoveResponse** message.
- 5.1.14. ONVIF Client invokes **Stop** request with parameters
  - VideoSource := *videoSourceToken1*
- 5.1.15. The DUT responds with **StopResponse** message.
- 5.1.16. ONVIF Client invokes **GetUsage** request with parameters
  - VideoSource := *videoSourceToken1*
- 5.1.17. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage3*

5.1.18. If *usage3.Pan* <= *usage2.Pan* item, FAIL the test and skip other steps.

**Test Result:****PASS –**

- DUT passes all assertions.

**FAIL –**

- DUT did not send **PanMoveResponse** message(s).
- DUT did not send **GetUsageResponse** message(s).
- DUT did not send **StopResponse** message(s).

## 5.1.2 PROVISIONING TILT MOVE AND STOP OPERATION

**Test Case ID:** PROVISIONING-1-1-2

**Specification Coverage:** Tilt Move (Provisioning Service Specification), Stop (Provisioning Service Specification)

**Feature Under Test:** TiltMove, Stop

**WSDL Reference:** provisioning.wsdl

**Test Purpose:** To verify Tilt movement and stopping using the DUT Provisioning TiltMove operation.

**Pre-Requisite:** Provisioning Service is received from the DUT.

**Test Configuration:** ONVIF Client and DUT

**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client retrieves Provisioning Service capabilities by following the procedure mentioned in [Annex A.1](#) with the following input and output parameters
  - out *cap* - Provisioning Service capabilities

4. Set *timeout1* := "P2D"
5. For each Source *cap.Source* in *cap* repeat the following steps:
  - 5.1. If *cap.Source.MaximumTiltMoves* is present:
    - 5.1.1. Set *videoSourceToken1* := *cap.Source.@VideoSourceToken*
    - 5.1.2. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.3. The DUT responds with **GetUsageResponse** with parameters
      - *Usage* := *usage1*
    - 5.1.4. If there is no *usage1.Tilt* item, FAIL the test and skip other steps.
    - 5.1.5. ONVIF Client invokes **TiltMove** request with parameters
      - *VideoSource* := *videoSourceToken1*
      - *Direction* := "Up"
      - Timeout skipped
    - 5.1.6. The DUT responds with **TiltMoveResponse** message.
    - 5.1.7. Set *defaultTimeout* := *cap.DefaultTimeout*
    - 5.1.8. Wait until *defaultTimeout* timeout expires.
    - 5.1.9. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.10. The DUT responds with **GetUsageResponse** with parameters
      - *Usage* := *usage2*
    - 5.1.11. If *usage2.Tilt* <= *usage1.Tilt* item, FAIL the test and skip other steps.
    - 5.1.12. ONVIF Client invokes **TiltMove** request with parameters
      - *VideoSource* := *videoSourceToken1*
      - *Direction* := "Down"
      - *Timeout* := ~~*timeout1*~~



5.1.13. The DUT responds with **TiltMoveResponse** message.

5.1.14. ONVIF Client invokes **Stop** request with parameters

- VideoSource := *videoSourceToken1*

5.1.15. The DUT responds with **StopResponse** message.

5.1.16. ONVIF Client invokes **GetUsage** request with parameters

- VideoSource := *videoSourceToken1*

5.1.17. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage3*

5.1.18. If *usage3.Tilt* <= *usage2.Tilt* item, FAIL the test and skip other steps.

#### Test Result:

##### PASS –

- DUT passes all assertions.

##### FAIL –

- DUT did not send **TiltMoveResponse** message(s).
- DUT did not send **GetUsageResponse** message(s).
- DUT did not send **StopResponse** message(s).

## 5.1.3 PROVISIONING ZOOM MOVE AND STOP OPERATION

**Test Case ID:** PROVISIONING-1-1-3

**Specification Coverage:** Zoom Move (Provisioning Service Specification), Stop (Provisioning Service Specification)

**Feature Under Test:** ZoomMove, Stop

**WSDL Reference:** provisioning.wsdl

**Test Purpose:** To verify Zoom movement and stopping using the DUT Provisioning ZoomMove operation.

**Pre-Requisite:** Provisioning Service is received from the DUT.

## Test Configuration: ONVIF Client and DUT

### Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client retrieves Provisioning Service capabilities by following the procedure mentioned in [Annex A.1](#) with the following input and output parameters
  - out *cap* - Provisioning Service capabilities
4. Set *timeout1* := "P2D"
5. For each Source *cap.Source* in *cap* repeat the following steps:
  - 5.1. If *cap.Source.MaximumZoomMoves* is present:
    - 5.1.1. Set *videoSourceToken1* := *cap.Source.@VideoSourceToken*
    - 5.1.2. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.3. The DUT responds with **GetUsageResponse** with parameters
      - *Usage* := *usage1*
    - 5.1.4. If there is no *usage1.Zoom* item, FAIL the test and skip other steps.
    - 5.1.5. ONVIF Client invokes **ZoomMove** request with parameters
      - *VideoSource* := *videoSourceToken1*
      - *Direction* := "Wide"
      - Timeout skipped
    - 5.1.6. The DUT responds with **ZoomMoveResponse** message.
    - 5.1.7. Set *defaultTimeout* := *cap.DefaultTimeout*
    - 5.1.8. Wait until *defaultTimeout* timeout expires.
    - 5.1.9. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.10. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage2*

5.1.11. If *usage2.Zoom* <= *usage1.Zoom* item, FAIL the test and skip other steps.

5.1.12. ONVIF Client invokes **ZoomMove** request with parameters

- VideoSource := *videoSourceToken1*
- Direction := "Telephoto"
- Timeout := *timeout1*

5.1.13. The DUT responds with **ZoomMoveResponse** message.

5.1.14. ONVIF Client invokes **Stop** request with parameters

- VideoSource := *videoSourceToken1*

5.1.15. The DUT responds with **StopResponse** message.

5.1.16. ONVIF Client invokes **GetUsage** request with parameters

- VideoSource := *videoSourceToken1*

5.1.17. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage3*

5.1.18. If *usage3.Zoom* <= *usage2.Zoom* item, FAIL the test and skip other steps.

#### Test Result:

##### PASS –

- DUT passed all assertions.

##### FAIL –

- DUT did not send **ZoomMoveResponse** message(s).
- DUT did not send **GetUsageResponse** message(s).
- DUT did not send **StopResponse** message(s).

## 5.1.4 PROVISIONING ROLL MOVE AND STOP OPERATION

Test Case ID: PROVISIONING-1-1-4

**Specification Coverage:** Roll Move (Provisioning Service Specification), Stop (Provisioning Service Specification)

**Feature Under Test:** RollMove, Stop

**WSDL Reference:** provisioning.wsdl

**Test Purpose:** To verify Roll movement and stopping using the DUT Provisioning RollMove operation.

**Pre-Requirement:** Provisioning Service is received from the DUT.

**Test Configuration:** ONVIF Client and DUT

**Test Procedure:**

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client retrieves Provisioning Service capabilities by following the procedure mentioned in [Annex A.1](#) with the following input and output parameters
  - out *cap* - Provisioning Service capabilities
4. Set *timeout1* := "P2D"
5. For each Source *cap*.Source in *cap* repeat the following steps:
  - 5.1. If *cap*.Source.MaximumRollMoves is present:
    - 5.1.1. Set *videoSourceToken1* := *cap*.Source.@VideoSourceToken
    - 5.1.2. ONVIF Client invokes **GetUsage** request with parameters
      - VideoSource := *videoSourceToken1*
    - 5.1.3. The DUT responds with **GetUsageResponse** with parameters
      - Usage := *usage1*
    - 5.1.4. If there is no *usage1*.Roll item, FAIL the test and skip other steps.
    - 5.1.5. ONVIF Client invokes **RollMove** request with parameters
      - VideoSource := *videoSourceToken1*
      - Direction := "Clockwise"
      - Timeout skipped

- 5.1.6. The DUT responds with **RollMoveResponse** message.
- 5.1.7. Set *defaultTimeout* := *cap.DefaultTimeout*
- 5.1.8. Wait until *defaultTimeout* timeout expires.
- 5.1.9. ONVIF Client invokes **GetUsage** request with parameters
  - *VideoSource* := *videoSourceToken1*
- 5.1.10. The DUT responds with **GetUsageResponse** with parameters
  - *Usage* := *usage2*
- 5.1.11. If *usage2.Roll* <= *usage1.Roll* item, FAIL the test and skip other steps.
- 5.1.12. If *cap.Source* supports *AutoLevel*:
  - 5.1.12.1 ONVIF Client invokes **RollMove** request with parameters
    - *VideoSource* := *videoSourceToken1*
    - *Direction* := "Auto"
    - *Timeout* := *timeout1*
  - 5.1.12.2 The DUT responds with **RollMoveResponse** message.
  - 5.1.12.3 Wait until *timeout1* Timeout expires.
  - 5.1.12.4 ONVIF Client invokes **GetUsage** request with parameters
    - *VideoSource* := *videoSourceToken1*
  - 5.1.12.5 The DUT responds with **GetUsageResponse** with parameters
    - *Usage* := *usage3*
  - 5.1.12.6 If *usage3.Roll* <= *usage2.Roll* item, FAIL the test and skip other steps.
- 5.1.13. ONVIF Client invokes **RollMove** request with parameters
  - *VideoSource* := *videoSourceToken1*
  - *Direction* := "Counterclockwise"
  - *Timeout* := *timeout1*

- 5.1.14. The DUT responds with **RollMoveResponse** message.
- 5.1.15. ONVIF Client invokes **Stop** request with parameters
- VideoSource := *videoSourceToken1*
- 5.1.16. The DUT responds with **StopResponse** message.
- 5.1.17. ONVIF Client invokes **GetUsage** request with parameters
- VideoSource := *videoSourceToken1*
- 5.1.18. The DUT responds with **GetUsageResponse** with parameters
- Usage := *usage4*
- 5.1.19. If *usage4.Roll* <= *usage3.Roll* item, FAIL the test and skip other steps.

**Test Result:****PASS –**

- DUT passes all assertions.

**FAIL –**

- DUT did not send **RollMoveResponse** message(s).
- DUT did not send **GetUsageResponse** message(s).
- DUT did not send **StopResponse** message(s).

## 5.1.5 PROVISIONING FOCUS MOVE AND STOP OPERATION

**Test Case ID:** PROVISIONING-1-1-5

**Specification Coverage:** Focus Move (Provisioning Service Specification), Stop (Provisioning Service Specification)

**Feature Under Test:** FocusMove, Stop

**WSDL Reference:** provisioning.wsdl

**Test Purpose:** To verify Focus movement and stopping using the DUT Provisioning FocusMove operation.

**Pre-Requisite:** Provisioning Service is received from the DUT.

## Test Configuration: ONVIF Client and DUT

### Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client retrieves Provisioning Service capabilities by following the procedure mentioned in [Annex A.1](#) with the following input and output parameters
  - out *cap* - Provisioning Service capabilities
4. Set *timeout1* := "P2D"
5. For each Source *cap.Source* in *cap* repeat the following steps:
  - 5.1. If *cap.Source.MaximumFocusMoves* is present:
    - 5.1.1. Set *videoSourceToken1* := *cap.Source.@VideoSourceToken*
    - 5.1.2. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.3. The DUT responds with **GetUsageResponse** with parameters
      - *Usage* := *usage1*
    - 5.1.4. If there is no *usage1.Focus* item, FAIL the test and skip other steps.
    - 5.1.5. ONVIF Client invokes **FocusMove** request with parameters
      - *VideoSource* := *videoSourceToken1*
      - *Direction* := "Near"
      - Timeout skipped
    - 5.1.6. The DUT responds with **FocusMoveResponse** message.
    - 5.1.7. Set *defaultTimeout* := *cap.DefaultTimeout*
    - 5.1.8. Wait until *defaultTimeout* timeout expires.
    - 5.1.9. ONVIF Client invokes **GetUsage** request with parameters
      - *VideoSource* := *videoSourceToken1*
    - 5.1.10. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage2*

5.1.11. If *usage2.Focus* <= *usage1.Focus* item, FAIL the test and skip other steps.

5.1.12. If *cap.Source* supports AutoFocus:

5.1.12.1 ONVIF Client invokes **FocusMove** request with parameters

- VideoSource := *videoSourceToken1*
- Direction := "Auto"
- Timeout := *timeout1*

5.1.12.2 The DUT responds with **FocusMoveResponse** message.

5.1.12.3 Wait until *timeout1* Timeout expires.

5.1.12.4 ONVIF Client invokes **GetUsage** request with parameters

- VideoSource := *videoSourceToken1*

5.1.12.5 The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage3*

5.1.12.6 If *usage3.Focus* <= *usage2.Focus* item, FAIL the test and skip other steps.

5.1.13. ONVIF Client invokes **FocusMove** request with parameters

- VideoSource := *videoSourceToken1*
- Direction := "Far"
- Timeout := *timeout1*

5.1.14. The DUT responds with **FocusMoveResponse** message.

5.1.15. ONVIF Client invokes **Stop** request with parameters

- VideoSource := *videoSourceToken1*

5.1.16. The DUT responds with **StopResponse** message.

5.1.17. ONVIF Client invokes **GetUsage** request with parameters

- VideoSource := *videoSourceToken1*



5.1.18. The DUT responds with **GetUsageResponse** with parameters

- Usage := *usage4*

5.1.19. If *usage4.Focus* <= *usage3.Focus* item, FAIL the test and skip other steps.

**Test Result:**

**PASS –**

- DUT passed all assertions.

**FAIL –**

- DUT did not send **RollMoveResponse** message(s).
- DUT did not send **GetUsageResponse** message(s).
- DUT did not send **StopResponse** message(s).

## 5.2 Capabilities

### 5.2.1 GET SERVICES AND GET PROVISIONING SERVICE CAPABILITIES CONSISTENCY

**Test Case ID:** PROVISIONING-2-1-1

**Specification Coverage:** Capability exchange (ONVIF Core Specification), Capabilities (Provisioning Service Specification)

**Feature Under Test:** GetServices, GetServiceCapabilities (Provisioning)

**WSDL Reference:** devicemgmt.wsdl, provisioning.wsdl

**Test Purpose:** To verify getting Provisioning Service using GetServices request. To verify Get Services and Provisioning Service Capabilities consistency.

**Pre-Requisite:** Provisioning Service is received from the DUT.

**Test Configuration:** ONVIF Client and DUT

**Test Procedure:**

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

3. ONVIF Client invokes **GetServices** message with parameters:
  - IncludeCapability := false
4. The DUT responds with a **GetServicesResponse** message with parameters:
  - Service list =: *listOfServicesWithoutCapabilities*
5. If *listOfServicesWithoutCapabilities* does not contain item with Namespace = "http://www.onvif.org/ver10/provisioning/wsd", FAIL the test and skip other steps.
6. Set *provisioningServ* := item from *listOfServicesWithoutCapabilities* list with Namespace = "http://www.onvif.org/ver10/provisioning/wsd".
7. If *provisioningServ.Capabilities* is specified, FAIL the test and skip other steps.
8. ONVIF Client invokes **GetServices** message with parameters:
  - IncludeCapability := true
9. The DUT responds with a **GetServicesResponse** message with parameters:
  - Service list =: *listOfServicesWithCapabilities*
10. If *listOfServicesWithCapabilities* does not contain item with Namespace = "http://www.onvif.org/ver10/provisioning/wsd", FAIL the test and skip other steps.
11. Set *provisioningServ* := item from *listOfServicesWithCapabilities* list with Namespace = "http://www.onvif.org/ver10/provisioning/wsd".
12. If *provisioningServ.Capabilities* is not specified, FAIL the test and skip other steps.
13. If *provisioningServ.Capabilities* does not contain valid Capabilities element for Provisioning service from "http://www.onvif.org/ver10/provisioning/wsd" namespace, FAIL the test and skip other steps.
14. ONVIF Client invokes **GetServiceCapabilities** (Provisioning) message.
15. The DUT responds with **GetServiceCapabilitiesResponse** message with parameters
  - Capabilities =: *cap*
16. If *cap* differs from *provisioningServ.Capabilities.Capabilities*, FAIL the test.

**Test Result:****PASS –**

- DUT passed all assertions.

**FAIL –**

- The DUT did not send **GetServicesResponse** message.
- The DUT did not send **GetServiceCapabilitiesResponse** message.

## Annex A Helper Procedures and Additional Notes

### A.1 Get Provisioning Service Capabilities

**Name:** HelperGetServiceCapabilities

**Procedure Purpose:** Helper procedure to get Provisioning Service Capabilities from the DUT.

**Pre-requisite:** Provisioning Service is received from the DUT.

**Input:** None

**Returns:** The service capabilities (*cap*).

**Procedure:**

1. ONVIF Client invokes **GetServiceCapabilities** request.
2. The DUT responds with **GetServiceCapabilitiesResponse** message with parameters
  - Capabilities =: *cap*

**Procedure Result:**

**PASS –**

- DUT passed all assertions.

**FAIL –**

- DUT did not send **GetServiceCapabilitiesResponse** message.