ONVIF^{TM}

Thermal Device Test Specification

Version 18.06

June 2018

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

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

REVISION HISTORY

Vers.	Date	Description
17.12	Aug 08, 2017	First Issue
18.06	Jun 21, 2018	Reformatting document using new template

Table of Contents

1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases need to verify the [ONVIF Thermal 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 Thermal Test Specification acts as a supplementary document to the [ONVIF Thermal 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 Thermal 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 Thermal Service which is defined in [ONVIF Thermal 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 Thermal Service Specs]; instead it would cover subset of it. The scope of this specification is to derive all the normative requirements of [ONVIF Thermal Service Specs] which are related to ONVIF Thermal Service and some of the optional requirements.

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

1.1.1 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 Thermal service address with GetServices command via Device service
- Getting capabilities with GetServiceCapabilities command
- Getting capabilities with GetServices command via Device service

1.1.2 Thermal Setting

Thermal Setting test cases are covered for verification to get and set thermal parameters for ColorPalette, NUCTable and Cooler. It means that **GetConfiguration**, **GetConfigurationOptions** and **SetConfiguration** commands are covered by this test case.

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 Thermal Service Specs] ONVIF Thermal Service Specification:

  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 Advanced Security Service and tests. For a list of applicable general terms and definitions, please see [ONVIF Base Test].

ColorPalette	Color Palette
Polarity	Polarity configuration
NUCTable	Non-Uniformity Correction (NUC) Table
Cooler	Cooler settings

4 Test Overview

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

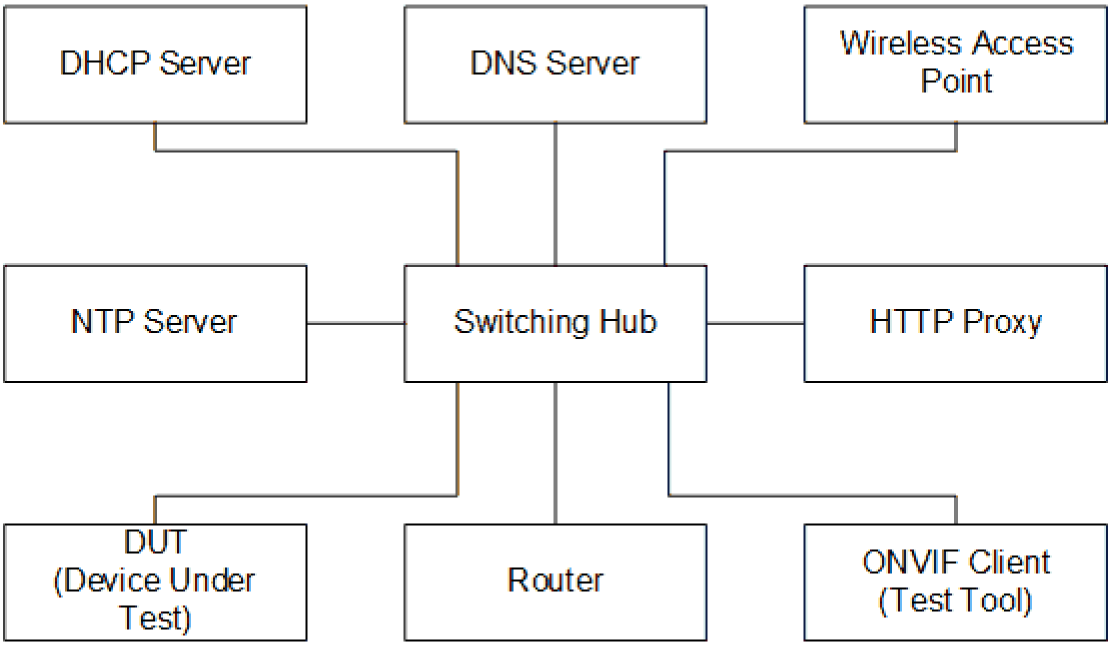
4.1 Test Setup

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 equipments in the test environment. All devices should be connected to the Switching Hub.

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 Capabilities

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

- DUT shall give the Thermal 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 5.1 for Capabilities Test Cases.

4.3.2 Thermal Setting

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

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

Please refer to Section 5.2 for Thermal Setting Test Cases.

5 Thermal Test Cases

5.1 Capabilities

5.1.1 Get Services and Get Thermal Service Capabilities Consistency

Test Case ID: THERMAL-1-1-1

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

Feature under test: GetServices, GetServiceCapabilities (Thermal)

WSDL Reference: devicemgmt.wsdl, thermal.wsdl

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

Pre-Requisite: Thermal Service was 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/thermal/wsdl", FAIL the test and skip other steps.

6. Set *thermalServ* := item from *listOfServicesWithoutCapabilities* list with Namespace = "http://www.onvif.org/ver10/thermal/wsdl".

7. If *thermalServ*.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/thermal/wsdl", FAIL the test and skip other steps.

11. Set *thermalServ* := item from *listOfServicesWithCapabilities* list with Namespace = "http://www.onvif.org/ver10/thermal/wsdl".

12. If *thermalServ*.Capabilities is not specified, FAIL the test and skip other steps.

13. If *thermalServ*.Capabilities does not contain valid Capabilities element for Thermal service from "http://www.onvif.org/ver10/thermal/wsdl" namespace, FAIL the test and skip other steps.

14. ONVIF Client invokes **GetServiceCapabilities** (Thermal) request.

15. The DUT responds with **GetServiceCapabilitiesResponse** message with parameters

    - Capabilities =: *cap*

16. If *cap* differs from *thermalServ*.Capabilities.Capabilities, FAIL the test.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **GetServicesResponse** messages.

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

5.2 Thermal Setting

5.2.1 Thermal Get Configuration and Get Configuration Options

Test Case ID: THERMAL-2-1-1

Specification Coverage: Get Configuration (Thermal Service Specification), Get Configuration Options (Thermal Service Specification)

Feature under test: GetConfiguration, GetConfigurationOptions

WSDL Reference: thermal.wsdl

Test Purpose: To verify retrieving of DUT Thermal Configuration using GetConfiguration command. To verify retrieving of DUT Thermal Configuration Options using GetConfigurationOptions command. To verify that all Configurations are consistent with Configuration Options.

Pre-Requisite: Thermal 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 Video Source Token list by following the procedure mentioned in Annex A.1 with the following input and output parameters

   - out *VideoSourceTokenList* - Video Source Token list

4. For each Video Source Token *videoSourceToken* in *VideoSourceTokenList* list repeat the following steps:

   4.1. ONVIF Client invokes **GetConfiguration** request with parameters

   - VideoSourceToken := *videoSourceToken*

   4.2. The DUT responds with **GetConfigurationResponse** message with parameters:

   - Configuration =: *configuration*

   4.3. ONVIF Client invokes **GetConfigurationOptions** request with parameters

   - VideoSourceToken := *videoSourceToken*

   4.4. The DUT responds with **GetConfigurationOptionsResponse** message with parameters:

   - ConfigurationOptions =: *configurationOptions*

   4.5. For each NUCTable (*nUCTable*) in *configurationOptions* repeat the following steps:

   - If *nUCTable*.LowTemperature is value greater than *nUCTable*.HighTemperature, FAIL the test and skip other steps.

   4.6. If *configuration*.ColorPalette is not equal to one of *configurationOptions*.ColorPalette items, FAIL the test and skip other steps.

4.7. If *configuration*.NUCTable is not equal to one of *configurationOptions*.NUCTable items, FAIL the test and skip other steps.

4.8. If *configurationOptions*.CoolerOptions.Enabled = false and *configuration*.Cooler.Enabled = false, FAIL the test and skip other steps.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- DUT did not send **GetConfigurationResponse** message.
- DUT did not send **GetConfigurationOptionsResponse** message.

Note: The following fields are compared at step 4.6:

- token
- Type
- Name

Note: The following fields are compared at step 4.7:

- token
- LowTemperature
- HighTemperature
- Name

5.2.2 Thermal Modify Configuration

Test Case ID: THERMAL-2-1-2

Specification Coverage: Set Configuration (Thermal Service Specification)

Feature under test: SetConfiguration

WSDL Reference: thermal.wsdl

Test Purpose: To verify modification of Thermal Configuration using SetConfiguration command.

Pre-Requisite: Thermal 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 Video Source Token list by following the procedure mentioned in Annex A.1 with the following input and output parameters

   - out *VideoSourceTokenList* - Video Source Token list

4. For each Video Source Token *videoSourceToken* in *VideoSourceTokenList* list repeat the following steps:

   4.1. ONVIF Client invokes **GetConfiguration** request with parameters

   - VideoSourceToken := *videoSourceToken*

   4.2. The DUT responds with **GetConfigurationResponse** message with parameters:

   - Configuration =: *initialConfiguration*

   4.3. ONVIF Client invokes **GetConfigurationOptions** request with parameters

   - VideoSourceToken := *videoSourceToken*

   4.4. The DUT responds with **GetConfigurationOptionsResponse** message with parameters:

   - ConfigurationOptions =: *configurationOptions1*

   4.5. If *configurationOptions1* contains at least two ColorPalette elements, do the following steps:

   4.5.1. ONVIF Client invokes **SetConfiguration** request with parameters:

   - VideoSourceToken =: *videoSourceToken*

   - Configuration.ColorPalette =: *configurationOptions1*.ColorPalette with @token other than *initialConfiguration*.ColorPalette.@token

   - Configuration.Polarity =: *initialConfiguration*.Polarity

   - Configuration.NUCTable =: *initialConfiguration*.NUCTable

   - Configuration.Cooler =: *initialConfiguration*.Cooler

4.5.2. The DUT responds with **SetConfigurationResponse** message.

4.5.3. ONVIF Client invokes **GetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*

4.5.4. The DUT responds with**GetConfigurationResponse** with parameters:

- Configuration =: *configuration2*

4.5.5. If *configuration2* is not equal to Configuration from step 4.5.1, FAIL the test and skip other steps.

4.5.6. ONVIF Client restores settings of Configuration with @token = *videoSourceToken*.

4.6. ONVIF Client invokes **SetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*
- Configuration.ColorPalette =: *initialConfiguration*.ColorPalette
- Polarity =: other value than *initialConfiguration*.@Polarity
- Configuration.NUCTable =: *initialConfiguration*.NUCTable
- Configuration.Cooler =: *initialConfiguration*.Cooler

4.7. The DUT responds with **SetConfigurationResponse** message.

4.8. ONVIF Client invokes **GetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*

4.9. The DUT responds with**GetConfigurationResponse** with parameters:

- Configuration =: *configuration2*

4.10. If *configuration2* is not equal to Configuration from step 4.6, FAIL the test and skip other steps.

4.11. ONVIF Client restores settings of Configuration with @token = *videoSourceToken*.

4.12. If *initialConfiguration* does not have NUCTable element and *configurationOptions1* contains NUCTable element or *configurationOptions1* contains at least two NUCTable elements, do the following steps:

4.12.1. ONVIF Client invokes **SetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*
- Configuration.ColorPalette =: *initialConfiguration*.ColorPalette
- Configuration.Polarity =: *initialConfiguration*.Polarity
- Configuration.NUCTable =: *configurationOptions1*.NUCTable with @token other than *initialConfiguration*.NUCTable.@token
- Configuration.Cooler =: *initialConfiguration*.Cooler

4.12.2. The DUT responds with **SetConfigurationResponse** message.

4.12.3. ONVIF Client invokes **GetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*

4.12.4. The DUT responds with**GetConfigurationResponse** with parameters:

- Configuration =: *configuration2*

4.12.5. If *configuration2* is not equal to Configuration from step 4.12.1, FAIL the test and skip other steps.

4.12.6. ONVIF Client restores settings of Configuration with @token = *videoSourceToken*.

4.13. If *configurationOptions1*.CoolerOptions = true, do the following steps:

4.13.1. ONVIF Client invokes **SetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*
- Configuration.ColorPalette =: *initialConfiguration*.ColorPalette
- Configuration.Polarity =: *initialConfiguration*.Polarity
- Configuration.NUCTable =: *initialConfiguration*.NUCTable
- Configuration.Cooler.Enabled =: other value than *initialConfiguration*.Cooler.Enabled

4.13.2. The DUT responds with **SetConfigurationResponse** message.

4.13.3. ONVIF Client invokes **GetConfiguration** request with parameters:

- VideoSourceToken =: *videoSourceToken*

4.13.4. The DUT responds with **GetConfigurationResponse** with parameters:

- Configuration =: *configuration2*

4.13.5. If *configuration2* is not equal to Configuration from step 4.13.1, FAIL the test and skip other steps.

4.13.6. ONVIF Client restores settings of Configuration with @token = *videoSourceToken*.

Test Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **GetConfigurationOptionsResponse** message.
- The DUT did not send **GetConfigurationResponse** message(s).
- The DUT did not send **SetConfigurationResponse** message(s).

Annex A Helper Procedures and Additional Notes

A.1 Get Video Source Token List

Name: HelperGetVideoSourceTokenList

Procedure Purpose: Helper procedure to retrieve Video Source Token List.

Pre-requisite: Thermal Service was received from the DUT.

Input: None.

Returns: Video Source Token list (*VideoSourceTokenList*).

Procedure:

1. ONVIF Client invokes **GetConfigurations** request.
2. The DUT responds with **GetConfigurationsResponse** message with parameters:
   - Configuration list =: *configurationsList*
3. If *configurationsList* is empty, FAIL the test and skip other steps.
4. For each Thermal Configuration *configuration* from *configurationsList* repeat the following steps
   - Add *configuration*.@token to *VideoSourceTokenList* list.

Procedure Result:

PASS –

- DUT passes all assertions.

FAIL –

- The DUT did not send **GetConfigurationsResponse** message.