# ONVIF®

# Profile C Client Test Specification

Version 19.06

June 2019

# REVISION HISTORY

| Vers. | Date | Description |
|---|---|---|
| 19.06 | Jun 14, 2019 | The following was done according to #309:<br><br>'Validated Feature' section for each feature updated to be synchronized with feature ID used in feature list.<br><br>'Feature Under Test' section for each test case updated to be synchronized with sub-feature ID used in feature list.<br><br>'Validated Feature List' test case section removed. |
| 18.06 | Jun 21, 2018 | Reformatting document using new template |
| 18.06 | Apr 05, 2018 | 'Required Number of Devices Summary' Annex added according to #241 |
| 18.06 | Feb 16, 2018 | The following were updated in the scope of #241:<br><br>Feature Level Requirement (updated with new rules)<br><br>Each Feature Level Requirement (updated with Check Condition based on Device Features and Required Number of Devices) |
| 17.06 | Jun 15, 2017 | Links in Normative references section were updated. |
| 16.07 | Apr 19, 2016 | • Test cases about specific event were removed: SYSTEMCOMPONENTSTATE-1, SYSTEMCOMPONENTSTATE-2, ACCESSCONTROLDECISIONS-1, ACCESSCONTROLDECISIONS-2, ACCESSCONTROLDECISIONS-3, ACCESSCONTROLDECISIONS-4, ACCESSCONTROLDECISIONS-5, ACCESSCONTROLDECISIONS-6, ACCESSCONTROLDECISIONS-7, ACCESSCONTROLDECISIONS-8, ACCESSCONTROLDECISIONS-9, CONFIGURATIONCHANGENOTIFICATION-1, CONFIGURATIONCHANGENOTIFICATION-2, CONFIGURATIONCHANGENOTIFICATION-3, CONFIGURATIONCHANGENOTIFICATION-4, CONFIGURATIONCHANGENOTIFICATION-5, CONFIGURATIONCHANGENOTIFICATION-6, DURESS-1.<br>• System Component State scenario updated<br>• Access Control Decisions scenario updated<br>• Configuration Change Notifications scenario updated<br>• Duress Notifications scenario updated |
| 16.07 | Apr 05, 2016 | The description about structure and hierarchy was replaced for the test cases: SYSTEMCOMPONENTINFORMATION-1, SYSTEMCOMPONENTINFORMATION-2, SYSTEMCOMPONENTINFORMATION-3, SYSTEMCOMPONENTSTATE-1, SYSTEMCOMPONENTSTATE-2, DOORCONTROL-1, DOORCONTROL-2, DOORCONTROL-3, DOORCONTROL-4, DOORCONTROL-5, DOORCONTROL-6, DOORCONTROL-7, ACCESSPOINTCONTROL-1, EXTERNALAUTHORIZATION-2<br><br>Old description: |

| | | |
|---|---|---|
| | | Client %COMMAND NAME% request message is a well-formed SOAP request (refer to onvif.xsd) AND<br><br>Client %COMMAND NAME% request message has a proper hierarchy (refer to %SERVICE%.wsdl) AND<br><br>New description:<br><br>Client %COMMAND NAME% request messages are valid according to XML Schemas listed in Namespaces AND<br><br>Client %COMMAND NAME% request in Test Procedure fulfills the following requirements:<br><br>The following steps was removed because the requirements are fullfield by XML Schemas validation:<br><br>• SYSTEMCOMPONENTSTATE-1:<br><br>[S6] "<PullMessages>" includes tag: "<Timeout>" AND<br><br>[S7] "<PullMessages>" includes tag: "<MessageLimit>" AND<br>• SYSTEMCOMPONENTSTATE-2:<br><br>[S6] "<PullMessages>" includes tag: "<Timeout>" AND<br><br>[S7] "<PullMessages>" includes tag: "<MessageLimit>" AND<br>• EXTERNALAUTHORIZATION-2:<br><br>[S3] "<ExternalAuthorization>" includes tag: "<Decision>" AND<br><br>[S4] "<Decision>" contains value EITHER ("Granted" OR "Denied") AND |
| 16.07 | Mar 14, 2016 | www.onvif.org was removed from Copyright section. |
| 16.01 | Dec 2, 2015 | General item (Test Owerview) was added<br><br>Minor updates in formatting, typos and terms according review result of other Client Test Specifications<br><br>EXTERNALAUTHORIZATION-3 was removed. Related feature was chnaged in accordance.<br><br>EXTERNALAUTHORIZATION-1 was updated to include new pre-requisite and new test style was upplied. |
| 16.01 | Nov 20, 2015 | Change according to # 67:Expected Scenarios Under Test of Access Control Decisions, Configuration Change Notifications, Duress Notifications were updated: dependence on Device features were added. New Note was added into corresponding test cases. |
| 16.01 | Sep 28, 2015 | Added Access Control Decisions Test Cases, Configuration Change Notifications, Duress Notifications Test Cases sections |
| 15.06 | Jun 10, 2015 | No major changes were made, just minor formatting fixes. |
| 15.05 | May 20, 2015 | No major changes were made, just minor grammatical corrections. |
| 15.03 | Mar 20, 2015 | Added External Authorization Test Cases section. |
| 15.02 | Feb 19, 2015 | Pass criteria in SYSTEMCOMPONENTSTATE-1 and 2 test cases have been updated (added additional criteria for checking <TopicExpression> tag value). |

| | | |
|---|---|---|
| 14.12 | Dec 11, 2014 | Fixed typos and inconsistencies. |
| 1.0 | Oct 16, 2014 | Initial version |

Table of Contents

# 1 Introduction

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

This particular document defines test cases required for testing Profile C features of a Client application e.g. System Component Information, System Component State, Door Control and Access Point Control. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

## 1.1 Scope

This ONVIF Profile C Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile C features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Profile C features according to ONVIF Profile Specifications.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Profile C features.

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

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

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

## 1.2 System Component Information

System Component Information section specifies Client ability to request lists of Access Points, Doors and Areas from Device.

## 1.3 System Component State

System Component State section specifies Client ability to request information about the state of Access Points (enabled/disabled) and Doors (locked, unlocked, etc.).

## 1.4 Door Control

Door Control section specifies Client ability to to control Doors (access door, lock door, unlock door, etc.).

## 1.5 Access Point Control

Access Point Control section specifies Client ability to control Access Points (enabled/disabled).

## 1.6 External Authorization

External Authorization section specifies Client ability to receive authorization request from Device and then make decisions about granting access and send it to Device. This section also specifies Client ability to retrieve and receive notifications about access decisions related to External Authorization.

## 1.7 Access Control Decisions

Access Control Decisions section specifies Client ability to receive notifications about access decisions related to Access Control.

## 1.8 Configuration Change Notifications

Configuration Change Notifications section specifies Client ability to receive notifications about access points, doors, and areas configuration change.

## 1.9 Duress Notifications

Duress Notifications section specifies Client ability to receive notifications about duress situation.

# 2 Normative references

- ONVIF Conformance Process Specification:

  https://www.onvif.org/profiles/conformance/

- ONVIF Profile Policy:

  https://www.onvif.org/profiles/

- ONVIF Core Specifications:

  https://www.onvif.org/profiles/specifications/

- ONVIF Core Client Test Specification:

  https://www.onvif.org/profiles/conformance/client-test/

- ONVIF Profile C Specification:

  https://www.onvif.org/profiles/profile-c/

- ONVIF Access Control Specification:

  https://www.onvif.org/profiles/specifications/

- ONVIF Door Control Specification:

  https://www.onvif.org/profiles/specifications/

- ISO/IEC Directives, Part 2, Annex H:

  http://www.iso.org/directives

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

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

# 3 Terms and Definitions

## 3.1 Conventions

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

## 3.2 Definitions

This section describes terms and definitions used in this document.

| | |
|---|---|
| **Profile** | See ONVIF Profile Policy. |
| **Profile C** | The Profile C Specification. |
| **ONVIF Device** | Computer appliance or software program that exposes one or multiple ONVIF Web Services. |
| **ONVIF Client** | Computer appliance or software program that uses ONVIF Web Services. |
| **Conversation** | A Conversation is all exchanges between two MAC addresses that contains SOAP request and response. |
| **Network** | A network is an interconnected group of devices communicating using the Internet protocol. |
| **Network Trace Capture file** | Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications. |
| **SOAP** | SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. |
| **Client Test Tool** | ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set. |
| **Door** | A physical door, barrier, turnstile, etc which can be controlled remotely and restricts access between two areas. A door is usually equipped with an electronic lock and a sensor. |
| **Door Alarm** | An abnormal state of the door where door is forced open or held open beyond the permitted time duration. |
| **Door Mode** | Logical state of the door indicating whether the door is locked, unlocked, blocked, locked down or locked open etc. |
| **Lock** | An operation after which a door is locked and alarm is unmasked. |
| **Unlock** | An operation to allow a door to be freely used for passage without any door alarms being triggered. |

| | |
|---|---|
| **Access Point** | A logical composition of a physical door and ID point(s) controlling access in one direction. |
| **Disable Access Point** | If an Access Point is disabled, it will not be considered in the decision making process and no commands will be issued from that Access Point to the Door configured for that Access Point. When an Access Point is disabled, the associated ID Point may or may not be disabled or shut down. Clients may still be able to command the Door Controller to control associated door even though that door is also referenced by a disabled access point. |
| **ID Point** | A device that converts reader signals to protocols recognized by an authorization engine. It can be card reader, REX, biometric reader etc. |
| **Valid Device Response** | Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared. |

## 3.3 Abbreviations

This section describes abbreviations used in this document.

**PACS** Physical Access Control System.

**tns1:** A prefix for the ONVIF topic namespace "http://www.onvif.org/ver10/topics". This prefix is not part of the standard and an implementation can use any prefix. See Core Specification description of Namespaces for details.

**HTTP** Hyper Text Transport Protocol.

**HTTPS** Hyper Text Transport Protocol over Secure Socket Layer.

**URI** Uniform Resource Identifier.

**WSDL** Web Services Description Language.

**XML** eXtensible Markup Language.

## 3.4 Namespaces

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

### Table 3.1. Defined namespaces in this specification

| Prefix | Namespace URI | Description |
|---|---|---|
| soapenv | http://www.w3.org/2003/05/soap-envelope | Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1] |
| xs | http://www.w3.org/2001/XMLSchema | Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2] |
| xsi | http://www.w3.org/2001/XMLSchema-instance | XML schema instance namespace |

| Prefix | Namespace URI | Description |
|---|---|---|
| tns1 | http://www.onvif.org/ver10/topics | The namespace for the ONVIF topic namespace |
| tt | http://www.onvif.org/ver10/schema | ONVIF XML schema descriptions |
| tev | http://www.onvif.org/ver10/events/wsdl | The namespace for the WSDL event service |
| tac | http://www.onvif.org/ver10/accesscontrol/wsdl | The namespace for the WSDL access control service |
| tdc | http://www.onvif.org/ver10/doorcontrol/wsdl | The namespace for the WSDL door control service |

# 4 Test Overview

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

An ONVIF Client conformant to Profile C is an ONVIF Client that can request information regarding the Physical Access Control System (PACS) related entities from an ONVIF Device conformant to Profile C and do basic control of Doors and Access Points over an IP network. ONVIF Client can also retrieve and receive standardized PACS related events.

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

## 4.1 General

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

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

### 4.1.1 Feature Level Requirement

Feature Level Requirement item contains a feature ID, check condition based on Device features, required number of Devices and feature requirement level for the Profiles, which will be used for Profiles conformance.

To claim this Feature as supported Client shall pass Expected Scenario Under Test:

- for each Device, which supports Device Features defined in Check Condition Based on Device Features
- for at least with number of Devices specified in Required Number of Devices

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall support this Feature to claim this Profile Conformance.

### 4.1.2 Expected Scenarios Under Test

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

### 4.1.3 Test Cases

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

Each Test Case contains the following parts:

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

## 4.2 Test Setup

Collect Network Traces files required by the test cases.

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

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

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

## 4.3 Prerequisites

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

The Device shall be configured with an IPv4 address.

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

# 5 System Component Information Test Cases

## 5.1 Feature Level Requirement:

**Validated Feature:** System Component Information (SystemComponentInformation)

**Check Condition based on Device Features:** Access Control Service and Door Control Service are supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 5.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a lists of Access Points, Doors and Areas.
2. Client is considered as supporting System Component Information if the following conditions are met:
   - Client is able to list available Access Points using GetAccessPointInfoList operation AND
   - Client is able to list available Doors using GetDoorInfoList operation AND
   - Client is able to list available Areas using GetAreaInfoList operation.
3. Client is considered as NOT supporting System Component Information if ANY of the following is TRUE:
   - No valid responses for GetAccessPointInfoList OR
   - No valid responses for GetDoorInfoList OR
   - No valid responses for GetAreaInfoList.

## 5.3 LISTING OF ACCESS POINTS

**Test Label:** System Component Information - Listing of Access Points

**Test Case ID:** SYSTEMCOMPONENTINFORMATION-1

**Feature Under Test:** Listing of Access Points (SystemComponentInformation_ListingOfAccessPoints)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that list of all access points items provided by Device is received by Client using the GetAccessPointInfoList operation.

**Pre-Requisite:**

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

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

1. Client invokes GetAccessPointInfoList request message to retrieve complete list of all access points configured on the Device.
2. Device responds with code HTTP 200 OK and GetAccessPointInfoListResponse message.

**Test Result:**

**PASS -**

- Client **GetAccessPointInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAccessPointInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetAccessPointInfoList>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetAccessPointInfoListResponse>" tag AND
  - [S4] At least one Device response in the same Conversation does not contain: "<NextStartReference>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 5.4 LISTING OF DOORS

**Test Label:** System Component Information - Listing of Doors

**Test Case ID:** SYSTEMCOMPONENTINFORMATION-2

**Feature Under Test:** Listing of Doors (SystemComponentInformation_ListingOfDoors)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that list of all doors items provided by Device is received by Client using the GetDoorInfoList operation.

**Pre-Requisite:**

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

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

1. Client invokes GetDoorInfoList request message to retrieve complete list of all doors configured on the Device.
2. Device responds with code HTTP 200 OK and GetDoorInfoListResponse message.

**Test Result:**

**PASS -**

- Client **GetDoorInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetDoorInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetDoorInfoList>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetDoorInfoListResponse>" tag AND
  - [S4] At least one Device response in the same Conversation does not contain: "<NextStartReference>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 5.5 LISTING OF AREAS

**Test Label:** System Component Information - Listing of Areas

**Test Case ID:** SYSTEMCOMPONENTINFORMATION-3

**Feature Under Test:** Listing of Areas (SystemComponentInformation_ListingOfAreas)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that list of all areas items provided by Device is received by Client using the GetAreaInfoList operation.

**Pre-Requisite:**

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

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

1. Client invokes GetAreaInfoList request message to retrieve complete list of all areas configured on the Device.
2. Device responds with code HTTP 200 OK and GetAreaInfoListResponse message.

**Test Result:**

**PASS -**

- Client **GetAreaInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAreaInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetAreaInfoList>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetAreaInfoListResponse>" tag AND
  - [S4] At least one Device response in the same Conversation does not contain: "<NextStartReference>" tag.

**FAIL -**

- The Client failed PASS criteria.

# 6 System Component State Test Cases

## 6.1 Feature Level Requirement:

**Validated Feature:** System Component State (SystemComponentState)

**Check Condition based on Device Features:** Access Control Service and Door Control Service are supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 6.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation to get notifications about the state of access points.
2. Client uses Pull Point event mechanism to retrieve notification events from Device.
3. Client is considered as supporting System Component State if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client supports SystemComponentInformation_AccessPointInfoList feature AND
   - Client supports SystemComponentInformation_DoorInfoList feature AND
   - Client is able to receive tns1:AccessPoint/State/Enabled notification about a state of access point if Device supports AccessPointStateEnabledEvent AND
   - Client is able to retrieve at least one of the following notifications about a state of door:
     - tns1:Door/State/DoorMode notification if Device supports DoorModeEvent
     - tns1:Door/State/DoorPhysicalState notification if Device supports DoorPhysicalStateEvent
     - tns1:Door/State/LockPhysicalState notification if Device supports LockPhysicalStateEvent
     - tns1:Door/State/DoubleLockPhysicalState notification if Device supports DoubleLockPhysicalStateEvent

- tns1:Door/State/DoorAlarm notification if Device supports DoorAlarmEvent
- tns1:Door/State/DoorTamper notification if Device supports DoorTamperEvent
- tns1:Door/State/DoorFault notification if Device supports DoorFaultEvent

4. Client is considered as NOT supporting System Component State if ANY of the following is TRUE:
   - Client does not support EventHandling_Pullpoint feature AND
   - Client does not support SystemComponentInformation_AccessPointInfoList feature AND
   - Client does not support SystemComponentInformation_DoorInfoList feature AND
   - Client is not able to receive tns1:AccessPoint/State/Enabled notification about a state of access point if Device supports AccessPointStateEnabledEvent AND
   - Client is not able to retrieve the following notifications about a state of door:
     - tns1:Door/State/DoorMode notification if Device supports DoorModeEvent
     - tns1:Door/State/DoorPhysicalState notification if Device supports DoorPhysicalStateEvent
     - tns1:Door/State/LockPhysicalState notification if Device supports LockPhysicalStateEvent
     - tns1:Door/State/DoubleLockPhysicalState notification if Device supports DoubleLockPhysicalStateEvent
     - tns1:Door/State/DoorAlarm notification if Device supports DoorAlarmEvent
     - tns1:Door/State/DoorTamper notification if Device supports DoorTamperEvent
     - tns1:Door/State/DoorFault notification if Device supports DoorFaultEvent

# 7 Door Control Test Cases

## 7.1 Feature Level Requirement:

**Validated Feature:** Door Control (DoorControl)

**Check Condition based on Device Features:** Door Control Service and Access Door and Lock Door and Unlock Door are supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 7.2 Expected Scenarios Under Test:

1. Client invokes a specific, valid mandatory Door Control command in order to change the state of door.
2. Client is considered as supporting Door Control if the following conditions are met:
    - Device returns a valid response to AccessDoor request AND
    - Device returns a valid response to LockDoor request AND
    - Device returns a valid response to UnlockDoor request
    - When Device and Client support any of the following conditional features:
        - Device returns a valid response to DoubleLockDoor request OR
        - Device returns a valid response to BlockDoor request
    - When Device and Client support LockDown conditional features:
        - Device returns a valid response to LockDownDoor request AND
        - Device returns a valid response to LockDownReleaseDoor request
    - When Device and Client support LockOpen conditional features:
        - Device returns a valid response to LockOpenDoor request AND
        - Device returns a valid response to LockOpenReleaseDoor request.
3. Client is considered as NOT supporting Door Control if ANY of the following is TRUE:

- No valid Device response to AccessDoor request OR
- No valid Device response to LockDoor request OR
- No valid Device response to UnlockDoor request
- When Device and Client support any of the following conditional features:
  - No valid Device response to DoubleLockDoor request AND
  - No valid Device response to BlockDoor request
- When Device and Client support LockDown conditional features:
  - No valid Device response to LockDownDoor request OR
  - No valid Device response to LockDownReleaseDoor request
- When Device and Client support LockOpen conditional features:
  - No valid Device response to LockOpenDoor request OR
  - No valid Device response to LockOpenReleaseDoor request.

## 7.3 ACCESS DOOR

**Test Label:** Door Control - AccessDoor

**Test Case ID:** DOORCONTROL-1

**Feature Under Test:** Access Door (DoorControl_AccessDoor)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that Client is able to change the state of door using AccessDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes AccessDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and AccessDoorResponse message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

## 7.4 LOCK DOOR

**Test Label:** Door Control - LockDoor

**Test Case ID:** DOORCONTROL-2

**Feature Under Test:** Lock Door (DoorControl_LockDoor)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that Client is able to change the state of door using LockDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes LockDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and LockDoorResponse message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

## 7.5 UNLOCK DOOR

**Test Label:** Door Control - UnlockDoor

**Test Case ID:** DOORCONTROL-3

**Feature Under Test:** Unlock Door (DoorControl_UnlockDoor)

**Profile C Normative Reference:** Mandatory

**Test Purpose:** To verify that Client is able to change the state of door using UnlockDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes UnlockDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and UnlockDoorResponse message.

**Test Result:**

**PASS -**

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

- Client **UnlockDoor** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<UnlockDoor>" tag after the "<Body>" tag AND
  - [S2] "<UnlockDoor>" includes tag: "<Token>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<UnlockDoorResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 7.6 DOUBLE LOCK DOOR

**Test Label:** Door Control - DoubleLockDoor

**Test Case ID:** DOORCONTROL-4

**Feature Under Test:** Double Lock Door (DoorControl_DoubleLockDoor)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to change the state of door using DoubleLockDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes DoubleLockDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and DoubleLockDoorResponse message.

**Test Result:**

**PASS -**

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

- [S1] Client request contains "<DoubleLockDoor>" tag after the "<Body>" tag AND
- [S2] "<DoubleLockDoor>" includes tag: "<Token>" with non-empty string value of specific token AND
- [S3] Device response contains "HTTP/* 200 OK" AND
- [S4] Device response contains "<DoubleLockDoorResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 7.7 BLOCK DOOR

**Test Label:** Door Control - BlockDoor

**Test Case ID:** DOORCONTROL-5

**Feature Under Test:** Block Door (DoorControl_BlockDoor)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to change the state of door using BlockDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes BlockDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and BlockDoorResponse message.

**Test Result:**

**PASS -**

- Client **BlockDoor** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **BlockDoor** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<BlockDoor>" tag after the "<Body>" tag AND

- [S2] "<BlockDoor>" includes tag: "<Token>" with non-empty string value of specific token AND
- [S3] Device response contains "HTTP/* 200 OK" AND
- [S4] Device response contains "<BlockDoorResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 7.8 LOCK DOWN DOOR

**Test Label:** Door Control - LockDownDoor

**Test Case ID:** DOORCONTROL-6

**Feature Under Test:** Lock Down Door (DoorControl_LockDownDoor)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to change the state of Door using LockDownDoor operation and then releasing this state using LockDownReleaseDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes LockDownDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and LockDownDoorResponse message.
3. Client invokes LockDownReleaseDoor request message to release the LockedDown state.
4. Device responds with code HTTP 200 OK and LockDownReleaseDoorResponse message.

**Test Result:**

**PASS -**

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

- [S1] Client request contains "<LockDownDoor>" tag after the "<Body>" tag AND
- [S2] "<LockDownDoor>" includes tag: "<Token>" with non-empty string value of specific token AND
- [S3] Device response contains "HTTP/* 200 OK" AND
- [S4] Device response contains "<LockDownDoorResponse>" tag AND

- Client **LockDownReleaseDoor** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **LockDownReleaseDoor** request in Test Procedure fulfills the following requirements:
  - [S5] Client request contains "<LockDownReleaseDoor>" tag after the "<Body>" tag AND
  - [S6] "<LockDownReleaseDoor>" includes tag: "<Token>" with token value from LockDownDoor operation AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<LockDownReleaseDoorResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

## 7.9 LOCK OPEN DOOR

**Test Label:** Door Control - LockOpenDoor

**Test Case ID:** DOORCONTROL-7

**Feature Under Test:** Lock Open Door (DoorControl_LockOpenDoor)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to change the state of Door using LockOpenDoor operation and then releasing this state using LockOpenReleaseDoor operation.

**Pre-Requisite:**

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

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

1. Client invokes LockOpenDoor request message to change the state of door.
2. Device responds with code HTTP 200 OK and LockOpenDoorResponse message.
3. Client invokes LockOpenReleaseDoor request message to release the LockOpenDoor state.
4. Device responds with code HTTP 200 OK and LockOpenReleaseDoorResponse message.

**Test Result:**

**PASS -**

- Client **LockOpenDoor** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **LockOpenDoor** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<LockOpenDoor>" tag after the "<Body>" tag AND
  - [S2] "<LockOpenDoor>" includes tag: "<Token>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<LockOpenDoorResponse>" tag. AND
- Client **LockOpenReleaseDoor** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **LockOpenReleaseDoor** request in Test Procedure fulfills the following requirements:
  - [S5] Client request contains "<LockOpenReleaseDoor>" tag after the "<Body>" tag AND
  - [S6] "<LockOpenReleaseDoor>" includes tag: "<Token>" with token value from LockOpenDoor operation AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<LockOpenReleaseDoorResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

# 8 Access Points Control Test Cases

## 8.1 Feature Level Requirement:

**Validated Feature:** Access Points Control (AccessPointControl)

**Check Condition based on Device Features:** Enable/Disable Access Point is supported by Device.

**Required Number of Devices:** 1

**Profile C Requirement:** Conditional

## 8.2 Expected Scenarios Under Test:

1. Client invokes a specific Access Points Control commands in order to change the state of access point.
2. Client is considered as supporting Access Points Control if the following conditions are met:
   - Device returns a valid response to EnableAccessPoint request AND
   - Device returns a valid response to DisableAccessPoint request.
3. Client is considered as NOT supporting Access Points Control if ANY of the following is TRUE:
   - No valid Device response to EnableAccessPoint request OR
   - No valid Device response to DisableAccessPoint request.

## 8.3 DISABLE ENABLE ACCESS POINT

**Test Label:** Access Points Control - DisableEnableAccessPoint

**Test Case ID:** ACCESSPOINTCONTROL-1

**Feature Under Test:** Disable Enable Access Point (AccessPointControl_DisableEnableAccessPoint)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to disable Access Point using DisableAccessPoint operation and enable Access Point using EnableAccessPoint operation.

**Pre-Requisite:**

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

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

1. Client invokes DisableAccessPoint request message to disable Access Point.
2. Device responds with code HTTP 200 OK and DisableAccessPointResponse message.
3. Client invokes EnableAccessPoint request message to enable access point.
4. Device responds with code HTTP 200 OK and EnableAccessPointResponse message.

**Test Result:**

**PASS -**

- Client **DisableAccessPoint** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DisableAccessPoint** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<DisableAccessPoint>" tag after the "<Body>" tag AND
  - [S2] "<DisableAccessPoint>" includes tag: "<Token>" with non-empty string value of specific token AND
  - [S3] Device response contains "HTTP/* 200 OK" AND
  - [S4] Device response contains "<DisableAccessPointResponse>" tag AND
- Client **EnableAccessPoint** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **EnableAccessPoint** request in Test Procedure fulfills the following requirements:
  - [S5] Client request contains "<EnableAccessPoint>" tag after the "<Body>" tag AND
  - [S6] "<EnableAccessPoint>" includes tag: "<Token>" with token value from DisableAccessPoint operation AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<EnableAccessPointResponse>" tag.

**FAIL -**

- The Client failed PASS criteria.

# 9 External Authorization Test Cases

## 9.1 Feature Level Requirement:

**Validated Feature:** External Authorization (ExternalAuthorization)

**Check Condition based on Device Features:** External Authorization is supported by Device.

**Required Number of Devices:** 1

**Profile C Requirement:** Conditional

## 9.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation.
2. Client receives authorization request from Device and makes a decision about granting access.
3. Client uses Pull Point event mechanism to retrieve notification events from Device.
4. Client receives notifications about access decisions related to External Authorization.
5. Client is considered as supporting External Authorization if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client supports access_control_decisions feature AND
   - Client is able to receive authorization request from Device AND
   - Client is able to send authorization decision to Device using **ExternalAuthorization** operation.
6. Client is considered as NOT supporting External Authorization if ANY of the following is TRUE:
   - Client does not support EventHandling_Pullpoint feature OR
   - Client does not support access_control_decisions feature OR
   - Client unable to receive authorization request from Device OR
   - No Valid Device Response to **ExternalAuthorization** request.

## 9.3 RECEIVE AUTHORIZATION REQUEST

**Test Label:** External Authorization - Receive Authorization Request

**Test Case ID:** EXTERNALAUTHORIZATION-1

**Feature Under Test:** Receive Authorization Request (ExternalAuthorization_ReceiveAuthRequest)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to receive authorization request from Device using Pull Point event mechanism.

**Pre-Requisite:**

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

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

1. Client invokes **CreatePullPointSubscription** message without any filter or with appropriate filter.
2. Device responds with code HTTP 200 OK and **CreatePullPointSubscriptionResponse** message.
3. Client invokes **PullMessages** command with Timeout and MessageLimit elements.
4. Device responds with code HTTP 200 OK and **PullMessagesResponse** message with corresponding event topic value.

**Test Result:**

**PASS -**

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

- If it contains **tev:Filter/wsnt:TopicExpression** with **Dialect** attribute equal to **http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete** then it fulfills the following requirements (else skip the check):
  - [S2] **wsnt:TopicExpression** element is equal to **tns1:AccessControl/Request/Credential** OR **tns1:AccessControl/Request/Anonymous** AND
- If it contains **tev:Filter/wsnt:TopicExpression** with **Dialect** attribute equal to **http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet** then it fulfills the following requirements (else skip the check):
  - [S3] **wsnt:TopicExpression** element contains **tns1:AccessControl/Request/Credential** OR **tns1:AccessControl/Request/Anonymous** OR **tns1:AccessControl/Request//.** OR **tns1:AccessControl//.** in expression AND

- Device response on the **CreatePullPointSubscription** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **tev:CreatePullPointSubscriptionResponse** AND
- Client **PullMessages** request in Test Procedure fulfills the following requirements:
  - [S6] **soapenv:Body** element has child element **tev:PullMessages** AND
- Device response on the **PullMessages** request fulfills the following requirements:
  - [S7] It has HTTP 200 response code AND
  - [S8] **soapenv:Body** element has child element **tev:PullMessagesResponse** AND
  - [S9] A least one **wsnt:NotificationMessage/wsnt:Topic** element has value equal to EITHER **tns1:AccessControl/Request/Credential** OR **tns1:AccessControl/Request/Anonymous**.

**FAIL -**

- The Client failed PASS criteria.

## 9.4 SEND AUTHORIZATION DECISION

**Test Label:** External Authorization - Send Authorization Decision

**Test Case ID:** EXTERNALAUTHORIZATION-2

**Feature Under Test:** Send Authorization Desicion (ExternalAuthorization_SendAuthDecision)

**Profile C Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to send Granted or Denied decision to Device.

**Pre-Requisite:**

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

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

1. Client sends ExternalAuthorization message to Device with Granted or Denied decision.
2. Device responds with code HTTP 200 OK and ExternalAuthorizationResponse message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

# 10 Access Control Decisions Test Cases

## 10.1 Feature Level Requirement:

**Validated Feature:** Access Control Decisions (AccessControlDecisions)

**Check Condition based on Device Features:** Access Control Service and tns1:AccessControl/AccessGranted/Credential and tns1:AccessControl/AccessDenied/Credential and tns1:AccessControl/AccessGranted/Anonymous and tns1:AccessControl/AccessDenied/AnonymousEvent tns1:AccessControl/AccessDenied/Credential/CredentialNotFoundCard and tns1:AccessControl/AccessTaken/Anonymous and tns1:AccessControl/AccessTaken/Credential and tns1:AccessControl/AccessNotTaken/Anonymous and tns1:AccessControl/AccessNotTaken/CredentialEvent are supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 10.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using CreatePullPointSubscription operation to get Access Control Decisions notifications.
2. Client is considered as supporting Access Control Decisions if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client supports SystemComponentInformation_AccessPointInfoList feature AND
   - Client is able to retrieve **tns1:AccessControl/AccessGranted/Credential** notification AND
   - Client is able to retrieve **tns1:AccessControl/Denied/Credential** notification AND
   - Client is able to retrieve **tns1:AccessControl/AccessGranted/Anonymous** notification AND
   - Client is able to retrieve **tns1:AccessControl/Denied/Anonymous** notification AND
   - Client is able to retrieve **tns1:AccessControl/Denied/CredentialNotFound/Card** notification AND
   - Client is able to retrieve **tns1:AccessControl/AccessTaken/Credential** notification AND

- Client is able to retrieve **tns1:AccessControl/AccessTaken/Anonymous** notification AND
- Client is able to retrieve **tns1:AccessControl/AccessNotTaken/Credential** notification AND
- Client is able to retrieve **tns1:AccessControl/AccessNotTaken/Anonymous** notification.

3. Client is considered as NOT supporting Access Control Decisions if ANY of the following is TRUE:
   - Client does not support EventHandling_Pullpoint feature OR
   - Client does not support SystemComponentInformation_AccessPointInfoList feature OR
   - Client unable to retrieve **tns1:AccessControl/AccessGranted/Credential** notification OR
   - Client unable to retrieve **tns1:AccessControl/Denied/Credential** notification OR
   - Client unable to retrieve **tns1:AccessControl/AccessGranted/Anonymous** notification OR
   - Client unable to retrieve **tns1:AccessControl/Denied/Anonymous** notification OR
   - Client unable to retrieve **tns1:AccessControl/Denied/CredentialNotFound/Card** notification OR
   - Client unable to retrieve **tns1:AccessControl/AccessTaken/Credential** notification OR
   - Client unable to retrieve **tns1:AccessControl/AccessTaken/Anonymous** notification OR
   - Client unable to retrieve **tns1:AccessControl/AccessNotTaken/Credential** notification OR
   - Client unable to retrieve **tns1:AccessControl/AccessNotTaken/Anonymous** notification.

# 11 Configuration Change Notifications Test Cases

## 11.1 Feature Level Requirement:

**Validated Feature:** Configuration Change Notifications (ConfigurationChangeNotifications)

**Check Condition based on Device Features:** tns1:AccessPoint/Changed and tns1:AccessPoint/Removed and tns1:Area/Changed and tns1:Area/Removed and tns1:Door/Changed and tns1:Door/Removed are supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 11.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation to get Configuration Change notifications.
2. Client uses Pull Point event mechanism to retrieve notification events from Device.
3. Client is considered as supporting Configuration change notification if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client supports SystemComponentInformation_AccessPointInfoList feature AND
   - Client supports SystemComponentInformation_DoorInfoList feature AND
   - Client supports SystemComponentInformation_AreaInfoList feature AND
   - Client is able to retrieve **tns1:Configuration/AccessPoint/Changed** notification AND
   - Client is able to retrieve **tns1:Configuration/AccessPoint/Removed** notification AND
   - Client is able to retrieve **tns1:Configuration/Door/Changed** notification AND
   - Client is able to retrieve **tns1:Configuration/Door/Removed** notification AND
   - Client is able to retrieve **tns1:Configuration/Area/Changed** notification AND
   - Client is able to retrieve **tns1:Configuration/Area/Removed** notification AND
4. Client is considered as NOT supporting Configuration change notification if ANY of the following is TRUE:

- Client does not support EventHandling_Pullpoint feature OR
- Client does not support SystemComponentInformation_AccessPointInfoList feature OR
- Client does not support SystemComponentInformation_DoorInfoList feature OR
- Client does not support SystemComponentInformation_AreaInfoList feature OR
- Client unable to retrieve **tns1:Configuration/AccessPoint/Changed** notification OR
- Client unable to retrieve **tns1:Configuration/AccessPoint/Removed** notification OR
- Client unable to retrieve **tns1:Configuration/Door/Changed** notification OR
- Client unable to retrieve **tns1:Configuration/Door/Removed** notification OR
- Client unable to retrieve **tns1:Configuration/Area/Changed** notification OR
- Client unable to retrieve **tns1:Configuration/Area/Removed** notification

# 12 Duress Notifications Test Cases

## 12.1 Feature Level Requirement:

**Validated Feature:** Duress Notifications (DuressNotifications)

**Check Condition based on Device Features:** Duress is supported by Device.

**Required Number of Devices:** 3

**Profile C Requirement:** Mandatory

## 12.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation.
2. Client uses Pull Point event mechanism to retrieve notification events from Device.
3. Client is considered as supporting Duress notification if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client supports SystemComponentInformation_AccessPointInfoList feature AND
   - Client is able to retrieve **tns1:AccessControl/Duress** notification.
4. Client is considered as NOT supporting Duress notification if ANY of the following is TRUE:
   - Client does not support EventHandling_Pullpoint feature OR
   - Client does not support SystemComponentInformation_AccessPointInfoList feature OR
   - Client unable to retrieve **tns1:AccessControl/Duress** notification.

# Annex A Test for Appendix A

## A.1 Required Number of Devices Summary

Required number of devices and Device feature dependency used in this test specification are listed in the Table.

### Table A.1. Required Number of Devices Summary

| Feature ID | Feature Name | Required Number of Devices | Check Condition based on Device Features | Check Condition based on Device Features ID |
|---|---|---|---|---|
| tc.SystemComponentInformation | System Component Information | 3 | Access Control Service and Door Control Service are supported by Device. | AccessControlService AND DoorControlService |
| tc.SystemComponentState | System Component State | 3 | Access Control Service and Door Control Service are supported by Device. | AccessControlService AND DoorControlService |
| tc.DoorControl | Door Control | 3 | Door Control Service and Access Door and Lock Door and Unlock Door are supported by Device. | DoorControlService AND AccessDoor AND LockDoor AND UnlockDoor |
| tc.AccessPointControl | Access Points Control | 1 | Enable/Disable Access Point is supported by Device. | EnableDisableAccessPoint |
| tc.ExternalAuthorization | External Authorization | 1 | External Authorization is supported by Device. | ExternalAuthorization |
| tc.AccessControlDecisions | Access Control Decisions | 3 | Check Condition based on Device Features: Access Control Service | AccessControlService AND AccessGrantedCredentialEv |

| Feature ID | Feature Name | Required Number of Devices | Check Condition based on Device Features | Check Condition based on Device Features ID |
|---|---|---|---|---|
| | | | and tns1:AccessControl/AccessGranted/Credential and tns1:AccessControl/AccessDenied/Credential and tns1:AccessControl/AccessGranted/Anonymous and tns1:AccessControl/AccessDenied/AnonymousEvent tns1:AccessControl/AccessDenied/Credential/CredentialNotFoundCard and tns1:AccessControl/AccessTaken/Anonymous and tns1:AccessControl/AccessTaken/Credential and tns1:AccessControl/AccessNotTaken/Anonymous and tns1:AccessControl/AccessNotTaken/CredentialEvent are supported by Device. | ent AND AccessDeniedCredentialEvent AND AccessGrantedAnonymousEvent AND AccessDeniedAnonymousEvent AccessDeniedCredentialCredentialNotFoundCardEvent AND AccessTakenAnonymousEvent AND AccessTakenCredentialEvent AND AccessNotTakenAnonymousEvent AND AccessNotTakenCredentialEvent |

| Feature ID | Feature Name | Required Number of Devices | Check Condition based on Device Features | Check Condition based on Device Features ID |
|---|---|---|---|---|
| tc.ConfigurationChangeNotifications | Configuration Change Notifications | 3 | tns1:AccessPoint/Changed and tns1:AccessPoint/Removed and tns1:Area/Changed and tns1:Area/Removed and tns1:Door/Changed and tns1:Door/Removed are supported by Device. | AccessPointChangedEvent AND AccessPointRemovedEvent AND AreaChangedEvent AND AreaRemovedEvent AND DoorChangedEvent AND DoorRemovedEvent |
| tc.DuressNotifications | Duress Notifications | 3 | Duress is supported by Device. | DuressEvent |