# ONVIF$^{TM}$

# Profile A Client Test Specification

Version 17.06

June 2017

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 |
|---|---|---|
| 17.06 | Jun 15, 2017 | Links in Normative references section were updated. |
| 16.12 | Dec 12, 2016 | • Test cases prefixes were changed from CONFIGURESPECIALDAYGROUP to CONFIGURESPECIALDAYGROUPS |
| 16.07 | Apr 18, 2016 | • Test cases about specific event were removed: CREDENTIALNOTIFICATIONS-1, CREDENTIALNOTIFICATIONS-2, CREDENTIALNOTIFICATIONS-3, SCHEDULENOTIFICATIONS-1, SCHEDULENOTIFICATIONS-2, ACCESSPROFILENOTIFICATIONS-1, ACCESSPROFILENOTIFICATIONS-2.<br>• Antipassback Violations Notifications Test Cases adedd<br>Special Days Notifications Test Cases adedd |
| 16.07 | Mar 24, 2016 | • get_credential_details feature was changed: Old description: "Client is able to get Credentials details using GetCredentials operation OR Client supports get_credential_list.get_credential_list feature" New description: "Client is able to get Credential details using GetCredentials operation"<br>• Get Access Profiles Details Test Cases adedd<br>Configure Access Profiles Test Cases adedd<br>Get Credential State Test Cases adedd<br>Change Credentials State Test Cases adedd<br>Get Schedules Details Test Cases adedd<br>Configure Schedules Test Cases added<br>Get Schedules State Test Cases added<br>Reset Antipassback Violation Test Cases added<br>Get Special Day Groups List Test Cases added<br>Get Special Day Groups Details Test Cases added<br>Configure Special Day Groups Test Cases added |
| 16.07 | Mar 14, 2016 | • www.onvif.org was removed from Copyright section. |
| 16.01 | Dec 07, 2016 | • General item (Test Overview) was added<br>• Minor updates in formatting, typos and terms<br>• Updates according review results (general changes): All test cases and use cases<br>• The following tests logic was updated to include logic for the case when all items were received in first GetXListResponse:<br>• GETCREDENTIALLIST-1<br>• GETCREDENTIALLIST-2<br>• GETSCHEDULELIST-1<br>• GETSCHEDULELIST-2<br>• GETACCESSPROFILELIST-1 |

| | | |
|---|---|---|
| | | • GETACCESSPROFILELIST-2 |
| 15.10 | Oct 13, 2016 | • Initial version:<br>General parts added<br>Get Credentials List Test Cases added<br>Get Credentials Details Test Cases added<br>Credential Configuration and State Notifications Test Cases added<br>Configure Credentials Test Cases<br>Get Schedules List Test Cases added<br>Schedules Configuration Notifications Test Cases added<br>Get Access Profiles List Test Cases added<br>Access Profiles Configuration Notifications Test Cases added<br>Get Credential Capabilities added |

## 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 A features of a Client application e.g. Get Credentials Capabilities, Get Credential List, Get Credential Details, Configure Credentials, Credential Configuration and State Notifications, Get Schedule List, Schedule Configuration Notifications, Get Access Profiles, Access Profile Configuration Notifications, Get Access Profile Details, Configure Access Profiles, Get Credential State, Change Credential State, Get Schedule Details, Configure Schedules, Get Schedule State, Reset Antipassback Violation, Get Special Day Group List, Get Special Day Group Details, Configure Special Day Groups. 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 A Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile A 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 A features according to ONVIF Profile Specifications.

The principal intended purposes are:

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

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

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

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

## 1.2 Get Credential Capabilities

Get Credential Capabilities section specifies Client ability to request lcapabilities of Credential Service from Device.

## 1.3 Get Credential List

Get Credential List section specifies Client ability to request lists of Credentials from Device.

## 1.4 Get Credential Details

Get Credentials Detail section specifies Client ability to request detailed information about Credentials.

## 1.5 Configure Credentials

Configure Credentials section specifies Client ability configure Credentials on Device.

## 1.6 Credential Configuration and State Notifications

Credential Configuration and State Notifications section specifies Client ability to receive from Device configuration and state notifications for Credentials.

## 1.7 Get Schedule List

Get Schedule List section specifies Client ability to request lists of Schedules from Device.

## 1.8 Schedule Configuration Notifications

Schedule Configuration Notifications section specifies Client ability to receive from Device configuration notifications for Schedules.

## 1.9 Get Access Profile List

Get Access Profile List section specifies Client ability to request lists of Access Profiles from Device.

## 1.10 Access Profile Configuration Notifications

Access Profile Configuration Notifications section specifies Client ability to receive from Device configuration notifications for Access Profiles.

## 1.11 Get Access Profile Details

Get Access Profile Details section specifies Client ability to request detailed information about Access Profiles.

## 1.12 Configure Access Profiles

Configure Access Profiles section specifies Client ability configure Access Profiles on Device.

## 1.13 Get Credential State

Get Credential State section specifies Client ability to get Credential state.

## 1.14 Change Credential State

Change Credential State section specifies Client ability to enable and disable Credential.

## 1.15 Get Schedule Details

Get Schedule Details section specifies Client ability to request detailed information about Schedules.

## 1.16 Configure Schedules

Configure Schedules section specifies Client ability configure Schedules on Device.

## 1.17 Get Schedule State

Get Schedule State section specifies Client ability to get schedule state.

## 1.18 Reset Antipassback Violation

Reset Antipassback Violation section specifies Client ability to reset antipassback violation for a specified credential.

## 1.19 Antipassback Violation Notifications Notifications

Antipassback Violation Notifications section specifies Client ability to receive from Device notifications about antipassback violation.

## 1.20 Get Special Day Group List

Get Special Day Group List section specifies Client ability to request lists of Special Day Groups from Device.

## 1.21 Get Special Day Group Details

Get Special Day Group Details section specifies Client ability to request detailed information about Special Day Groups.

## 1.22 Configure Special Day Groups

Configure Special Day Groups section specifies Client ability configure Special Day Groups on Device.

## 1.23 Special Days Notifications

Special Days Notifications section specifies Client ability to receive from Device configuration notifications for Special Days.

# 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 A Specification:

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

- ONVIF Access Rules Specification:

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

- ONVIF Credential Specification:

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

- ONVIF Schedule 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

- WS-BaseNotification:

  http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

- W3C SOAP 1.2, Part 1, Messaging Framework:

  http://www.w3.org/TR/soap12-part1/

- W3C XML Schema Part 1: Structures Second Edition:

  http://www.w3.org/TR/xmlschema-1/

- W3C XML Schema Part 2: Datatypes Second Edition:

  "http://www.w3.org/TR/xmlschema-2/ [http://www.w3.org/TR/xmlschema-2/]

# 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 A** The Profile A 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.

**Access Policy** An association of an access point and a schedule. An access policy defines when an access point can be accessed using an access profile which contains this access policy.

**Access Profile** A collection of access policies, used to define role based access.

**Access Point** A logical composition of a physical door and ID point(s) controlling access in one direction.

**Credential** A physical/tangible object, a piece of knowledge, or a facet of a person's physical being, that enables an individual access to a given physical facility or computer-based information system.

**Validity Period** From a certain point in time, to a later point in time.

**Schedule** A set of time periods, for example: working hours (weekdays from 08:00 AM to 06:00 PM). It may also include one or more special days schedule.

**ID Point** A device that converts reader signals to protocols recognized by an authorization engine. It can be card reader, REX, biometric reader etc.

**Anti-Passback** Operating mode which requires user validation when leaving a security controlled area in order to be able to re-enter and vice versa.

**Anti-Passback Violation State** A signal stating if the anti-passback rules have been violated for a credential.

**Credential Format** The credential data can be formatted in many different ways. ONVIF supports the BACnet format types in [ISO 16484-5:2014-09 Annex P].

**Credential Holder** Associates a credential with a user. Typically it holds a reference to a credential and a reference to a user.

**Credential Identifier** Card number, unique card information, PIN, fingerprint, or other biometric information, etc., that can be validated in an access point.

**Credential Number** A sequence of bytes uniquely identifying a credential at an access point.

**Credential State** The credential state indicates if a credential is enabled or disabled. The state also indicates if anti-passback has been violated or not. The state may also contain a reason why the credential was disabled.

**Duress** Forcing a person to provide access to a secure area against that person's wishes.

**Format Type** See Credential Format.

**iCalendar** An industry standard format for exchanging scheduling and activity-recording information electronically.

**Special Days** A set of dates that require the regular Schedule to be overridden, e.g. holidays, half-days or working Sundays.

**Special Days Schedule** A schedule that defines time periods for a Special Day List.

**Time Period** A time period has a start time and an end time, e.g. 8 AM to 6 PM.

**vEvent** A component in iCalendar, specifying the properties of an event.

**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. |
| **HTTP** | Hyper Text Transport Protocol. |

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

**URI** Uniform Resource Identifier.

**WSDL** Web Services Description Language.

**XML** eXtensible Markup Language.

## 3.4 Namespaces

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

### Table 3.1. Defined namespaces in this specification

| Prefix | Namespace URI | Description |
|---|---|---|
| soapenv | http://www.w3.org/2003/05/soap-envelope | Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1] |
| xs | http://www.w3.org/2001/XMLSchema | Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2] |
| xsi | http://www.w3.org/2001/XMLSchema-instance | XML schema instance namespace |
| tns1 | http://www.onvif.org/ver10/topics | The namespace for the ONVIF topic namespace |
| tt | http://www.onvif.org/ver10/schema | ONVIF XML schema descriptions |
| tds | http://www.onvif.org/ver10/device/wsdl | The namespace for the WSDL device service |
| 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 |
| tas | http://www.onvif.org/ver10/advancedsecurity/wsdl | The namespace for the WSDL advanced security service |
| tar | http://www.onvif.org/ver10/accessrules/wsdl | The namespace for the WSDL access rules service |
| tcr | http://www.onvif.org/ver10/credential/wsdl | The namespace for the WSDL credential service |
| tsc | http://www.onvif.org/ver10/schedule/wsdl | The namespace for the WSDL schedule service |
| wsnt | http://docs.oasis-open.org/wsn/b-2 | Schema namespace of the [WS-BaseNotification] specification. |

# 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 compliant to PACS Profile A can provide configurations of access rules, credentials and schedules. The 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 Normative Reference
- Expected Scenarios Under Test
- List of Test Cases

### 4.1.1 Feature Level Normative Reference

Feature Level Normative Reference item contains a feature ID and feature requirement level for the Profiles, which will be used for Profiles conformance.

If Feature Level Normative Reference is defined as Mandatory for some Profile, Client shall pass Expected Scenario Under Test for each Device with this Profile support to claim this Profile Conformance.

If Feature Level Normative Reference is defined as Conditional, Optional for some Profile, Client shall pass Expected Scenario Under Test for at least one Device with this Profile support to claim feature as supported.

### 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 - Normative Reference 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.
- Validated Feature List - list of features ID related to this test case.

## 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 A, 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 Get Credential Capabilities Test Cases

## 5.1 Feature Level Normative Reference:

**Validated Feature:** GetCredentialCapabilities

**Profile A Requirement:** Mandatory

## 5.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a credential service capabilities.
2. Client is considered as supporting Get Credential Capabilities if the following conditions are met:
   - Client is able to retrieve a credential service capabilities using **GetServiceCapabilities** operation (Credential Service) OR supports get_services_capabilities.get_services feature (Please, see ONVIF Core Client Test Specification).
3. Client is considered as NOT supporting Get Credential Capabilities if ANY of the following is TRUE:
   - No valid response **GetServiceCapabilities** request (Credential Service) AND get_credential_capabilities.get_services feature is not supported by Client.

## 5.3 GET SERVICE CAPABILITIES

**Test Label:** Get Credential Capabilities - Get Service Capabilities

**Test Case ID:** GETCREDENTIALCAPABILITIES-1

**Profile A Normative Reference:** Optional

**Feature Under Test:** Get Services

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

**Pre-Requisite:**

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

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

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

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_credential_capabilities.get_service_capabilities

# 6 Get Credential List Test Cases

## 6.1 Feature Level Normative Reference:

**Validated Feature:** GetCredentialList

**Profile A Requirement:** Mandatory

## 6.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Credentials.
2. Client is considered as supporting Get Credential List if the following conditions are met:
   - Client is able to list available Credentials using **GetCredentialInfoList** operation OR **GetCredentialList** operation.
3. Client is considered as NOT supporting Get Credential List if ANY of the following is TRUE:
   - No valid responses for **GetCredentialInfoList** request OR **GetCredentialList** request OR
   - **GetCredentialInfoList** request contains **tcr:StartReference** element value that was not recieved in **GetCredentialInfoList** response in **tcr:NextStartReference** element OR
   - **GetCredentialList** request contains **tcr:StartReference** element value that was not recieved in **GetCredentialList** response in **tcr:NextStartReference** element OR
   - Complete Credentials list was not received.

## 6.3 LISTING OF CREDENTIALS

**Test Label:** Get Credential List - Listing of Credentials

**Test Case ID:** GETCREDENTIALLIST-1

**Profile A Normative Reference:** Optional

**Feature Under Test:** Get Credential List

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

**Pre-Requisite:**

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

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

1. Client invokes **GetCredentialList** request message with skipped **tcr:StartReference** element to retrieve first part of the list of all credentials configured on the Device.
2. Device responds with code HTTP 200 OK and **GetCredentialListResponse** message.
3. If **GetCredentialListResponse** message contains **tcr:NextStartReference** element Client invokes **GetCredentialList** request message with **tcr:StartReference** element equal to value of **tcr:NextStartReference** element to retrieve next part of the list of all credentials configured on the Device.
4. Client repeats the previous step while **GetCredentialListResponse** message contains **tcr:NextStartReference** element.

**Test Result:**

**PASS -**

- Client **GetCredentialList** request messages are valid according to XML Schemas listed in Namespaces AND
- First Client **GetCredentialList** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tcr:GetCredentialList** AND
  - [S2] It does not contain **tcr:StartReference** element AND
- If response on previous **GetCredentialList** request contains **tcr:NextStartReference** element each next Client **GetCredentialList** requests in Test Procedure fulfills the following requirements (else skip the checks):
  - [S3] **soapenv:Body** element has child element **tcr:GetCredentialList** AND
  - [S4] It contains **tcr:StartReference** element equal to **tcr:NextStartReference** element from response on previous **GetCredentialList** request AND
- Device responses on the each **GetCredentialList** request in Test Procedure fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tcr:GetCredentialListResponse** AND

- The last in Test Procedure Device response on **GetCredentialList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_credential_list.get_credential_list

## 6.4 LISTING OF CREDENTIAL INFO

**Test Label:** Get Credential List - Listing of Credential Info

**Test Case ID:** GETCREDENTIALLIST-2

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Get Credential Info List

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

**Pre-Requisite:**

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

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

1. Client invokes **GetCredentialInfoList** request message with skipped **tcr:StartReference** element to retrieve first part of the list of all credentials configured on the Device.
2. Device responds with code HTTP 200 OK and **GetCredentialInfoListResponse** message.
3. If **GetCredentialInfoListResponse** message contains **tcr:NextStartReference** element Client invokes **GetCredentialInfoList** request message with **tcr:StartReference** element equal to value of **tcr:NextStartReference** element to retrieve next part of the list of all credentials configured on the Device.
4. Client repeats the previous step while **GetCredentialInfoListResponse** message contains **tcr:NextStartReference** element.

**Test Result:**

**PASS -**

- Client **GetCredentialInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- First Client **GetCredentialInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tcr:GetCredentialInfoList** AND
  - [S2] It does not contain **tcr:StartReference** element AND
- If response on previous **GetCredentialInfoList** request contains **tcr:NextStartReference** element each next Client **GetCredentialInfoList** requests in Test Procedure fulfills the following requirements (else skip the checks):
  - [S3] **soapenv:Body** element has child element **tcr:GetCredentialInfoList** AND
  - [S4] It contains **tcr:StartReference** element equal to **tcr:NextStartReference** AND element from response on previous **GetCredentialInfoList** request AND
- Device responses on the each **GetCredentialInfoList** request in Test Procedure fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tcr:GetCredentialInfoListResponse** AND
- The last in Test Procedure Device response on **GetCredentialInfoList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_credential_list.get_credential_info_list

# 7 Get Credential Details Test Cases

## 7.1 Feature Level Normative Reference:

**Validated Feature:** GetCredentialDetails

**Profile A Requirement:** Mandatory

## 7.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a Credentials details.

2. Client is considered as supporting Get Credential Details if the following conditions are met:

   - Client is able to get Credential details using **GetCredentials** operation.

3. Client is considered as NOT supporting Get Credential Details if ANY of the following is TRUE:

   - No valid responses for **GetCredentials** request with at least one Credential listed in it.

## 7.3 GET CREDENTIALS

**Test Label:** Get Credential Details - Get Credentials

**Test Case ID:** GETCREDENTIALDETAILS-1

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Get Credentials

**Test Purpose:** To verify that credential details provided by Device is received by Client using the **GetCredentials** operation.

**Pre-Requisite:**

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

- Device supports Credential Service.

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

1. Client invokes **GetCredentials** request message to retrieve credential details for specified credentials from the Device.

2. Device responds with code HTTP 200 OK and **GetCredentialsResponse** message which contains at least one **tcr:Credential** element.

**Test Result:**

**PASS -**

- Client **GetCredentials** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetCredentials** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tcr:GetCredentials** AND
- Device response on the **GetCredentials** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tcr:GetCredentialsResponse** AND
  - [S4] It contains at least one **tcr:Credential** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_credential_details.get_credentials

# 8 Configure Credentials Test Cases

## 8.1 Feature Level Normative Reference:

**Validated Feature:** ConfigureCredentials

**Profile A Requirement:** Mandatory

## 8.2 Expected Scenarios Under Test:

1. Client supports get_credential_capabilities feature.

2. Client get supported format types of a specified identifier type using **GetSupportedFormatTypes** operation to use it for **CreateCredential** operation and **ModifyCredential** operation.

3. Client creates credentials on a Device using **CreateCredential** operation.

4. Client modifys credentials on a Device using **ModifyCredential** operation.

5. Client deletes credentials from a Device using **DeleteCredential** operation.

6. Client is considered as supporting Configure Credentials if the following conditions are met:

    - Client is able to get supported identifier types using **GetServiceCapabilities** operation or **GetServices** operation AND

    - Client is able to get supported format types of a specified identifier type using **GetSupportedFormatTypes** operation AND

    - Client is able to create credential using **CreateCredential** operation AND

    - Client is able to modify credential using **ModifyCredential** operation AND

    - Client is able to delete credential using **DeleteCredential** operation.

7. Client is considered as NOT supporting Configure Credentials if ANY of the following is TRUE:

    - No valid responses for **GetSupportedFormatTypes** request OR

    - No valid responses for **CreateCredential** request OR

    - No valid responses for **ModifyCredential** request OR

    - No valid responses for **DeleteCredential** request.

## 8.3 GET SUPPORTED FORMAT TYPES

**Test Label:** Configure Credentials - Get Supported Format Types

**Test Case ID:** CONFIGURECREDENTIALS-1

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Get Supported Format Types

**Test Purpose:** To verify that Client is able to get supported format types from Device for specified identifier type using the **GetSupportedFormatTypes** operation.

**Pre-Requisite:**

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

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

1. Client invokes **GetSupportedFormatTypes** request message to get supported format types from Device for specified identifier type.
2. Device responds with code HTTP 200 OK and **GetSupportedFormatTypesResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_credentials.get_supported_format_types

## 8.4 CREATE CREDENTIAL

**Test Label:** Configure Credentials - Create Credential

**Test Case ID:** CONFIGURECREDENTIALS-2

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Create Credential

**Test Purpose:** To verify that Client is able to create credential on Device using the **CreateCredential** operation.

**Pre-Requisite:**

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

- Device supports Credential Service.

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

1. Client invokes **GetSupportedFormatTypes** request message to get supported format types from Device for specified identifier type.

2. Device responds with code HTTP 200 OK and **GetSupportedFormatTypesResponse** message.

3. Client invokes **CreateCredential** request message to create credential on Device with identifier type from **GetSupportedFormatTypes** request message and format type from **GetSupportedFormatTypes** response message.

4. Device responds with code HTTP 200 OK and **CreateCredentialResponse** message.

**Test Result:**

**PASS -**

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

- Client **CreateCredential** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child **tcr:CreateCredential** element AND
  - [S2] **tcr:Credential/@token** attribute is empty (has empty string value) AND
  - [S3] IF it contains **tcr:Credential/tcr:ValidFrom** element AND **tcr:Credential/tcr:ValidTo** element THEN **tcr:ValidFrom** element value is less or equal to **tcr:ValidTo** element value AND
  - [S4] IF there is at least one **tcr:Credential/tcr:CredentialAccessProfile** element with child elements **tcr:ValidFrom** AND **tcr:ValidTo** THEN for all such **tcr:Credential/tcr:CredentialAccessProfile** elements **tcr:ValidFrom** element value is less or equal to **tcr:ValidTo** element value AND
- Device response on the **CreateCredential** request fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tcr:CreateCredentialResponse** AND
- For each **tcr:Credential/tcr:CredentialIdentifier** from the **CreateCredential** request in Test Procedure fulfills the following requirements:
  - There is a Client **GetSupportedFormatTypes** request in Test Procedure fulfills the following requirements:
    - [S7] It invoked for the same Device as for the Client **CreateCredential** request AND
    - [S8] It invoked before the Client **CreateCredential** request AND
    - [S9] **tcr:CredentialIdentifierTypeName** element value is equal to **tcr:Credential/tcr:CredentialIdentifier/tcr:Type/tcr:Name** element from the **CreateCredential** request AND
  - Device response on the **GetSupportedFormatTypes** request fulfills the following requirements:
    - [S10] It has HTTP 200 response code AND
    - [S11] There is **tcr:FormatTypeInfo/tcr:FormatType** element which value is equal to **tcr:Credential/tcr:CredentialIdentifier/tcr:Type/tcr:FormatType** element value for the corresponding **tcr:Credential/tcr:CredentialIdentifier** element from the **CreateCredential** request with **tcr:Credential/tcr:CredentialIdentifier/tcr:Type/tcr:Name** element value equal to **tcr:CredentialIdentifierTypeName** element value from the **GetSupportedFormatTypes** request.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_credentials.create_credential

## 8.5 MODIFY CREDENTIAL

**Test Label:** Configure Credentials - Modify Credential

**Test Case ID:** CONFIGURECREDENTIALS-3

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Modify Credential

**Test Purpose:** To verify that Client is able to modify credential on Device using the **ModifyCredential** operation.

**Pre-Requisite:**

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

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

1. Client invokes **ModifyCredential** request message to create credential on Device.
2. Device responds with code HTTP 200 OK and **ModifyCredentialResponse** message.

**Test Result:**

**PASS -**

- Client **ModifyCredential** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **ModifyCredential** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child **tcr:ModifyCredential** element AND
  - If it contains **tcr:Credential/tcr:ValidFrom** element AND **tcr:Credential/tcr:ValidTo** element then it fulfills the following requirements (else skip the checks):
    - [S2] **tcr:Credential/tcr:ValidFrom** element value is less or equal to **tcr:Credential/tcr:ValidTo** element value AND

- If it contains at least one **tcr:Credential/tcr:CredentialAccessProfile** with child elements **tcr:ValidFrom** AND **tcr:ValidTo** then it fulfills the following requirements (else skip the checks):
  - [S3] For all **tcr:Credential/tcr:CredentialAccessProfile** elements with child elements **tcr:ValidFrom** AND **tcr:ValidTotcr:ValidFrom** element value is less or equal to **tcr:ValidTo** element value AND

- Device response on the **ModifyCredential** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **tcr:ModifyCredentialResponse**.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_credentials.modify_credential

## 8.6 DELETE CREDENTIAL

**Test Label:** Configure Credentials - Delete Credential

**Test Case ID:** CONFIGURECREDENTIALS-4

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Delete Credential

**Test Purpose:** To verify that Client is able to delete credential from Device using the **DeleteCredential** operation.

**Pre-Requisite:**

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

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

1. Client invokes **DeleteCredential** request message to delete credential from the Device for specified credential.
2. Device responds with code HTTP 200 OK and **DeleteCredentialResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_credentials.delete_credential

# 9 Credential Configuration and State Notifications Test Cases

## 9.1 Feature Level Normative Reference:

**Validated Feature:** CredentialsNotifications

**Profile A Requirement:** Mandatory

## 9.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get credentials configuration notifications.

2. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get credential state notifications.

3. Client uses Pull Point event mechanism OR Basic Notification event mechanism (if Device supports Basic Notification event mechanism) to retrieve notification events from Device.

4. Client is considered as supporting Credential Configuration and State Notifications if the following conditions are met:

   - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature AND

   - Client supports get_credential_list feature AND

   - Client is able to retrieve tns1:Configuration/Credential/Changed notifications about credential configuration change AND

   - Client is able to retrieve tns1:Configuration/Credential/Removed notifications about credential removing AND

   - Client is able to retrieve tns1:Credential/State/Enabled notifications about credential enable state change.

5. Client is considered as NOT supporting Credential Configuration and State Notifications if ANY of the following is TRUE:

   - Client does not support EventHandling_Pullpoint feature AND EventHandling_WS-BaseNotification feature OR

   - Client does not support get_credential_list feature OR

- Client is not able to retrieve tns1:Configuration/Credential/Changed notifications about credential configuration change OR
- Client is not able to retrieve tns1:Configuration/Credential/Removed notifications about credential removing OR
- Client is not able to retrieve tns1:Credential/State/Enabled notifications about credential enable state change.

# 10 Get Schedule List Test Cases

## 10.1 Feature Level Normative Reference:

**Validated Feature:** GetScheduleList

**Profile A Requirement:** Mandatory

## 10.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Schedules.
2. Client is considered as supporting Get Schedule List if the following conditions are met:
   - Client is able to list available Schedules using **GetScheduleInfoList** operation OR **GetScheduleList** operation.
3. Client is considered as NOT supporting Get Schedule List if ANY of the following is TRUE:
   - No valid responses for **GetScheduleInfoList** request OR **GetScheduleList** request OR
   - **GetScheduleInfoList** request contains **tsc:StartReference** element value that was not recieved in **GetScheduleInfoList** response in **tsc:NextStartReference** element OR
   - **GetScheduleList** request contains **tsc:StartReference** element value that was not recieved in **GetScheduleList** response in **tsc:NextStartReference** element OR
   - Complete Schedules list was not received.

## 10.3 LISTING OF SCHEDULES

**Test Label:** Get Schedule List - Listing of Schedules

**Test Case ID:** GETSCHEDULELIST-1

**Profile A Normative Reference:** Optional

**Feature Under Test:** Get Schedule List

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

**Pre-Requisite:**

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

- Device supports Schedule Service.

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

1. Client invokes **GetScheduleList** request message with skipped **tsc:StartReference** element to retrieve first part of the list of all schedules configured on the Device.

2. Device responds with code HTTP 200 OK and **GetScheduleListResponse** message.

3. If **GetScheduleListResponse** message contains **tsc:NextStartReference** element Client invokes **GetScheduleList** request message with **tsc:StartReference** element equal to value of **tsc:NextStartReference** element to retrieve next part of the list of all schedules configured on the Device.

4. Client repeats the previous step while **GetScheduleListResponse** message contains **tsc:NextStartReference** element.

**Test Result:**

**PASS -**

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

- First Client **GetScheduleList** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tsc:GetScheduleList** AND

  - [S2] It does not contain **tsc:StartReference** element AND

- If response on previous **GetScheduleList** request contains **tcr:NextStartReference** element each next Client **GetScheduleList** requests in Test Procedure fulfills the following requirements (else skip the checks):

  - [S3] **soapenv:Body** element has child element **tsc:GetScheduleList** AND

  - [S4] It contains **tsc:StartReference** element equal to **tsc:NextStartReference** element from response on previous **GetScheduleList** request AND

- Device responses on the each **GetScheduleList** request in Test Procedure fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

  - [S6] **soapenv:Body** element has child element **tsc:GetScheduleListResponse** AND

- The last in Test Procedure Device response on **GetScheduleList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_schedule_list.get_schedule_list

## 10.4 LISTING OF SCHEDULE INFO

**Test Label:** Get Schedule List - Listing of Schedule Info

**Test Case ID:** GETSCHEDULELIST-2

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Get Schedule Info List

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

**Pre-Requisite:**

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

- Device supports Schedule Service.

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

1. Client invokes **GetScheduleInfoList** request message with skipped **tsc:StartReference** element to retrieve first part of the list of all schedules configured on the Device.

2. Device responds with code HTTP 200 OK and **GetScheduleInfoListResponse** message.

3. If **GetScheduleInfoListResponse** message contains **tsc:NextStartReference** element Client invokes **GetScheduleInfoList** request message with **tsc:StartReference** element equal to value of **tsc:NextStartReference** element to retrieve next part of the list of all schedules configured on the Device.

4. Client repeats the previous step while **GetScheduleInfoListResponse** message contains **tsc:NextStartReference** element.

**Test Result:**

**PASS -**

- Client **GetScheduleInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- First Client **GetScheduleInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tsc:GetScheduleInfoList** AND
  - [S2] It does not contain **tsc:StartReference** element AND
- If response on previous **GetScheduleInfoList** request contains **tcr:NextStartReference** element each next Client **GetScheduleInfoList** requests in Test Procedure fulfills the following requirements (else skip the checks):
  - [S3] **soapenv:Body** element has child element **tsc:GetScheduleInfoList** AND
  - [S4] It contains **tsc:StartReference** element equal to **tsc:NextStartReference** element from response on previous **GetScheduleInfoList** request AND
- Device responses on the each **GetScheduleInfoList** request in Test Procedure fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tsc:GetScheduleInfoListResponse** AND
- The last in Test Procedure Device response on **GetScheduleInfoList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_schedule_list.get_schedule_info_list

# 11 Schedule Configuration Notifications Test Cases

## 11.1 Feature Level Normative Reference:

**Validated Feature:** SchedulesNotifications

**Profile A Requirement:** Mandatory

## 11.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get schedules configuration notifications.
2. Client uses Pull Point event mechanism OR Basic Notification event mechanism (if Device supports Basic Notification event mechanism) to retrieve notification events from Device.
3. Client is considered as supporting Schedule Configuration Notifications if the following conditions are met:
    - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature AND
    - Client supports get_schedule_list feature AND
    - Client is able to retrieve tns1:Configuration/Schedule/Changed notifications about schedule configuration change AND
    - Client is able to retrieve tns1:Configuration/Schedule/Removed notifications about schedule removing AND
4. Client is considered as NOT supporting Schedule Configuration Notifications if ANY of the following is TRUE:
    - Client does not support EventHandling_Pullpoint feature AND EventHandling_WS-BaseNotification feature OR
    - Client does not support get_schedule_list feature OR
    - Client is not able to retrieve tns1:Configuration/Schedule/Changed notifications about schedule configuration change OR
    - Client is not able to retrieve tns1:Configuration/Schedule/Removed notifications about schedule removing.

# 12 Get Access Profile List Test Cases

## 12.1 Feature Level Normative Reference:

**Validated Feature:** GetAccessProfileList

**Profile A Requirement:** Mandatory

## 12.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Access Profiles.
2. Client is considered as supporting Get Access Profile List if the following conditions are met:
    - Client is able to list available Access Profiles using **GetAccessProfileInfoList** operation OR **GetAccessProfileList** operation.
3. Client is considered as NOT supporting Get Access Profile List if ANY of the following is TRUE:
    - No valid responses for **GetAccessProfileInfoList** request OR **GetAccessProfileList** request OR
    - **GetAccessProfileInfoList** request contains **tsc:StartReference** element value that was not recieved in **GetAccessProfileInfoList** response in **tsc:NextStartReference** element OR
    - **GetAccessProfileList** request contains **tsc:StartReference** element value that was not recieved in **GetAccessProfileList** response in **tsc:NextStartReference** element OR
    - Complete Access Profiles list was not received.

## 12.3 LISTING OF ACCESS PROFILES

**Test Label:** Get Access Profile List - Listing of Access Profiles

**Test Case ID:** GETACCESSPROFILELIST-1

**Profile A Normative Reference:** Optional

**Feature Under Test:** Get Access Profile List

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

**Pre-Requisite:**

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

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

1. Client invokes **GetAccessProfileList** request message with skipped **tar:StartReference** element to retrieve first part of the list of all access profiles configured on the Device.
2. Device responds with code HTTP 200 OK and **GetAccessProfileListResponse** message.
3. If **GetAccessProfileListResponse** message contains **tar:NextStartReference** element Client invokes **GetAccessProfileList** request message with **tar:StartReference** element equal to value of **tar:NextStartReference** element to retrieve next part of the list of all access profiles configured on the Device.
4. Client repeats the previous step while **GetAccessProfileListResponse** message contains **tar:NextStartReference** element.

**Test Result:**

**PASS -**

- Client **GetAccessProfileList** request messages are valid according to XML Schemas listed in Namespaces AND
- First Client **GetAccessProfileList** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tar:GetAccessProfileList** AND
  - [S2] It does not contain **tar:StartReference** element AND
- If response on previous **GetAccessProfileList** request contains **tcr:NextStartReference** element each next Client **GetAccessProfileList** requests in Test Procedure fulfills the following requirements (else skip the checks):
  - [S3] **soapenv:Body** element has child element **tar:GetAccessProfileList** AND
  - [S4] It contains **tar:StartReference** element equal to **tar:NextStartReference** element from response on previous **GetAccessProfileList** request AND
- Device responses on the each **GetAccessProfileList** request in Test Procedure fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tar:GetAccessProfileListResponse** AND
- The last in Test Procedure Device response on **GetAccessProfileList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_access_profile_list.get_access_profile_list

## 12.4 LISTING OF ACCESSPROFILE INFO

**Test Label:** Get Access Profile List - Listing of Access Profile Info

**Test Case ID:** GETACCESSPROFILELIST-2

**Profile A Normative Reference:** Mandatory

**Feature Under Test:** Get Access Profile Info List

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

**Pre-Requisite:**

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

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

1. Client invokes **GetAccessProfileInfoList** request message with skipped **tar:StartReference** element to retrieve first part of the list of all access profiles configured on the Device.
2. Device responds with code HTTP 200 OK and **GetAccessProfileInfoListResponse** message.
3. If **GetAccessProfileInfoListResponse** message contains **tar:NextStartReference** element Client invokes **GetAccessProfileInfoList** request message with

**tar:StartReference** element equal to value of **tar:NextStartReference** element to retrieve next part of the list of all access profiles configured on the Device.

4. Client repeats the previous step while **GetAccessProfileInfoListResponse** message contains **tar:NextStartReference** element.

**Test Result:**

**PASS -**

- Client **GetAccessProfileInfoList** request messages are valid according to XML Schemas listed in Namespaces AND
- First Client **GetAccessProfileInfoList** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tar:GetAccessProfileInfoList** AND
  - [S2] It does not contain **tar:StartReference** element AND
- If response on previous **GetAccessProfileInfoList** request contains **tcr:NextStartReference** element each next Client **GetAccessProfileInfoList** requests in Test Procedure fulfills the following requirements (else skip the checks):
  - [S3] **soapenv:Body** element has child element **tar:GetAccessProfileInfoList** AND
  - [S4] It contains **tar:StartReference** element equal to **tar:NextStartReference** element from response on previous **GetAccessProfileInfoList** request AND
- Device responses on the each **GetAccessProfileInfoList** request in Test Procedure fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] **soapenv:Body** element has child element **tar:GetAccessProfileInfoListResponse** AND
- The last in Test Procedure Device response on **GetAccessProfileInfoList** request fulfills the following requirements:
  - [S7] It does not contain **tcr:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_access_profile_list.get_access_profile_info_list

# 13 Access Profile Configuration Notifications Test Cases

## 13.1 Feature Level Normative Reference:

**Validated Feature:** AccessProfileNotifications

**Profile A Requirement:** Mandatory

## 13.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get access profiles configuration notifications.

2. Client uses Pull Point event mechanism OR Basic Notification event mechanism (if Device supports Basic Notification event mechanism) to retrieve notification events from Device.

3. Client is considered as supporting Access Profile Configuration if the following conditions are met:

    - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature AND

    - Client supports get_access_profile_list feature AND

    - Client is able to retrieve tns1:Configuration/AccessProfile/Changed notifications about access profile configuration change AND

    - Client is able to retrieve tns1:Configuration/AccessProfile/Removed notifications about access profile removing AND

4. Client is considered as NOT supporting Access Profile Configuration and State Notifications if ANY of the following is TRUE:

    - Client does not support EventHandling_Pullpoint feature AND EventHandling_WS-BaseNotification feature OR

    - Client does not support get_access_profile_list feature OR

    - Client is not able to retrieve tns1:Configuration/AccessProfile/Changed notifications about access profile configuration change OR

    - Client is not able to retrieve tns1:Configuration/AccessProfile/Removed notifications about access profile removing.

# 14 Get Access Profile Details Test Cases

## 14.1 Feature Level Normative Reference:

**Validated Feature:** GetAccessProfileDetails

**Profile A Requirement:** Conditional

## 14.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve Access Profiles details.
2. Client is considered as supporting Get Access Profiles Details if the following conditions are met:
   - Client is able to get Access Profiles details using **GetAccessProfiles** operation.
3. Client is considered as NOT supporting Get Access Profiles Details if ANY of the following is TRUE:
   - No valid responses for **GetAccessProfiles** request with at least one Access Profile listed in it.

## 14.3 GET ACCESS PROFILES

**Test Label:** Get Access Profiles Details - Get Access Profiles

**Test Case ID:** GETACCESSPROFILESDETAILS-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Access Profiles

**Test Purpose:** To verify that Client is able to get access profiles detailes using the **GetAccessProfiles** operation.

**Pre-Requisite:**

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

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

1. Client invokes **GetAccessProfiles** request message to get access profiles detailes from Device.

2. Device responds with code HTTP 200 OK and **GetAccessProfilesResponse** message which contains at least one **AccessProfile** element.

**Test Result:**

**PASS -**

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

- Client **GetAccessProfiles** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tar:GetAccessProfiles** AND

- Device response on the **GetAccessProfiles** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **tar:GetAccessProfilesResponse** AND

  - [S4] **tar:GetAccessProfilesResponse** has at least one **tar:AccessProfile** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_access_profile_details.get_access_profiles

# 15 Configure Access Profiles Test Cases

## 15.1 Feature Level Normative Reference:

**Validated Feature:** ConfigureAccessProfiles

**Profile A Requirement:** Conditional

## 15.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve access profiles details using **GetAccessProfiles** operation.

2. Client creates access profile on a Device using **CreateAccessProfile** operation.

3. Client modifys access profile on a Device using **ModifyAccessProfile** operation.

4. Client deletes access profile from a Device using **DeleteAccessProfile** operation.

5. Client is considered as supporting Configure Access Profiles if the following conditions are met:

   - Client is able to get access profiles details using **GetAccessProfiles** operation AND

   - Client is able to create access profile using **CreateAccessProfile** operation AND

   - Client is able to modify access profile using **ModifyAccessProfile** operation AND

   - Client is able to delete access profile using **DeleteAccessProfile** operation.

6. Client is considered as NOT supporting Configure Access Profiles if ANY of the following is TRUE:

   - No valid responses for **GetAccessProfiles** request with at least one Access Profile listed in it OR

   - No valid responses for **CreateAccessProfile** request OR

   - No valid responses for **ModifyAccessProfile** request OR

   - No valid responses for **DeleteAccessProfile** request.

## 15.3 CREATE ACCESS PROFILE

**Test Label:** Configure Access Profiles - Create Access Profile

**Test Case ID:** CONFIGUREACCESSPROFILES-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Create Access Profile

**Test Purpose:** To verify that Client is able to create access profile on Device using the **CreateAccessProfile** operation.

**Pre-Requisite:**

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

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

1. Client invokes **CreateAccessProfile** request message to create access profile on Device.
2. Device responds with code HTTP 200 OK and **CreateAccessProfileResponse** message.

**Test Result:**

**PASS -**

- Client **CreateAccessProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateAccessProfile** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child **tar:CreateAccessProfile** element AND
  - [S2] **tar:AccessProfile/@token** attribute is empty (has empty string value) AND
- Device response on the **CreateAccessProfile** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tar:CreateAccessProfileResponse**.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_access_profiles.create_access_profile

## 15.4 MODIFY ACCESS PROFILE

**Test Label:** Configure Access Profiles - Modify Access Profile

**Test Case ID:** CONFIGUREACCESSPROFILES-2

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Modify Access Profile

**Test Purpose:** To verify that Client is able to modify access profile on Device using the **ModifyAccessProfile** operation.

**Pre-Requisite:**

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

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

1. Client invokes **ModifyAccessProfile** request message to modify access profile on Device.
2. Device responds with code HTTP 200 OK and **ModifyAccessProfileResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_access_profiles.modify_access_profile

## 15.5 DELETE ACCESS PROFILE

**Test Label:** Configure Access Profiles - Delete Access Profile

**Test Case ID:** CONFIGUREACCESSPROFILES-3

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Delete Access Profile

**Test Purpose:** To verify that Client is able to delete access profile from Device using the **DeleteAccessProfile** operation.

**Pre-Requisite:**

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

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

1. Client invokes **DeleteAccessProfile** request message to delete access profile from the Device for specified access profile.
2. Device responds with code HTTP 200 OK and **DeleteAccessProfileResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_access_profiles.delete_access_profile

# 16 Get Credential State Test Cases

## 16.1 Feature Level Normative Reference:

**Validated Feature:** GetCredentialState

**Profile A Requirement:** Conditional

## 16.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a Credential state using **GetCredentialState** operation.

2. Client is considered as supporting Get Credential State if the following conditions are met:

   - Client is able to get Credential state using **GetCredentialState** operation AND

3. Client is considered as NOT supporting Get Credential State if ANY of the following is TRUE:

   - No valid responses for **GetCredentialState** request.

## 16.3 GET CREDENTIAL STATE

**Test Label:** Get Credential Sate

**Test Case ID:** GETCREDENTIALSTATE-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Credential State

**Test Purpose:** To verify that credential state provided by Device is received by Client using the **GetCredentialState** operation.

**Pre-Requisite:**

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

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

1. Client invokes **GetCredentialState** request message to retrieve credential state for specified credential from the Device.

2. Device responds with code HTTP 200 OK and **GetCredentialStateResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_credential_state.get_credential_state

# 17 Change Credential State Test Cases

## 17.1 Feature Level Normative Reference:

**Validated Feature:** ChangeCredentialState

**Profile A Requirement:** Conditional

## 17.2 Expected Scenarios Under Test:

1. Client connects to Device to change Credentials state using **EnableCredential** operation and **DisableCredential** operation.
2. Client is considered as supporting Change Credential State if the following conditions are met:
   - Client is able to change Credential state using **EnableCredential** operation AND
   - Client is able to change Credential state using **DisableCredential** operation AND
3. Client is considered as NOT supporting Change Credential State if ANY of the following is TRUE:
   - No valid responses for **EnableCredential** request OR
   - No valid responses for **DisableCredential** request.

## 17.3 ENABLE CREDENTIAL

**Test Label:** Change Credential State - Enable Credential

**Test Case ID:** CHANGECREDENTIALSTATE-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Enable Credential

**Test Purpose:** To verify that Client is able to change a credential state using the **EnableCredential** operation.

**Pre-Requisite:**

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

- Device supports Credential Service.

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

1. Client invokes **EnableCredential** request message to enable specified credential on a Device.
2. Device responds with code HTTP 200 OK and **EnableCredentialResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** change_credential_state.enable_credential

## 17.4 DISABLE CREDENTIAL

**Test Label:** Change Credential State - Disable Credential

**Test Case ID:** CHANGECREDENTIALSTATE-2

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Disable Credential

**Test Purpose:** To verify that Client is able to change a credential state using the **DisableCredential** operation.

**Pre-Requisite:**

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

- Device supports Credential Service.

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

1. Client invokes **DisableCredential** request message to enable specified credential on a Device.
2. Device responds with code HTTP 200 OK and **DisableCredentialResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** change_credential_state.disable_credential

# 18 Get Schedule Details Test Cases

## 18.1 Feature Level Normative Reference:

**Validated Feature:** GetScheduleDetails

**Profile A Requirement:** Conditional

## 18.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a Schedule details.

2. Client is considered as supporting Get Schedule Details if the following conditions are met:

   - Client is able to get schedule details using **GetSchedules** operation.

3. Client is considered as NOT supporting Get Schedule Details if ANY of the following is TRUE:

   - No valid responses for **GetSchedules** request with at least one schedule listed in it.

## 18.3 GET SCHEDULES

**Test Label:** Get Schedule Details - Get Schedules

**Test Case ID:** GETSCHEDULEDETAILS-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Schedules

**Test Purpose:** To verify that Client is able to get schedules detailes using the **GetSchedules** operation.

**Pre-Requisite:**

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

- Device supports Schedule Service.

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

1. Client invokes **GetSchedules** request message to get schedules detailes from Device.

2. Device responds with code HTTP 200 OK and **GetSchedulesResponse** message which contains at least one **Schedule** element.

**Test Result:**

**PASS -**

- Client **GetSchedules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetSchedules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tsc:GetSchedules** AND
- Device response on the **GetSchedules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tar:GetSchedulesResponse** AND
  - [S4] **tsc:GetSchedulesResponse** has at least one **tsc:Schedule** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_schedule_details.get_schedules

# 19 Configure Schedules Test Cases

## 19.1 Feature Level Normative Reference:

**Validated Feature:** ConfigureSchedules

**Profile A Requirement:** Conditional

## 19.2 Expected Scenarios Under Test:

1. Client creates schedule on a Device using **CreateSchedule** operation.
2. Client modifys schedule on a Device using **ModifySchedule** operation.
3. Client deletes schedule from a Device using **DeleteSchedule** operation.
4. Client is considered as supporting Configure Schedules if the following conditions are met:
   - Client is able to create schedule using **CreateSchedule** operation AND
   - Client is able to modify schedule using **ModifySchedule** operation AND
   - Client is able to delete schedule using **DeleteSchedule** operation.
5. Client is considered as NOT supporting Configure Schedules if ANY of the following is TRUE:
   - No valid responses for **CreateSchedule** request OR
   - No valid responses for **ModifySchedule** request OR
   - No valid responses for **DeleteSchedule** request.

## 19.3 CREATE SCHEDULE

**Test Label:** Configure Schedules - Create Schedule

**Test Case ID:** CONFIGURESCHEDULES-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Create Schedule

**Test Purpose:** To verify that Client is able to create schedule on Device using the **CreateSchedule** operation.

**Pre-Requisite:**

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

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

1. Client invokes **CreateSchedule** request message to create schedule on Device.
2. Device responds with code HTTP 200 OK and **CreateScheduleResponse** message.

**Test Result:**

**PASS -**

- Client **CreateSchedule** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateSchedule** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child **tsc:CreateSchedule** element AND
  - [S2] **tsc:Schedule/@token** attribute is empty (has empty string value) AND
- Device response on the **CreateSchedule** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tsc:CreateScheduleResponse**.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_schedules.create_schedule

## 19.4 MODIFY SCHEDULE

**Test Label:** Configure Schedules - Modify Schedule

**Test Case ID:** CONFIGURESCHEDULES-2

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Modify Schedule

**Test Purpose:** To verify that Client is able to modify schedule on Device using the **ModifySchedule** operation.

**Pre-Requisite:**

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

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

1. Client invokes **ModifySchedule** request message to modify schedule on Device.
2. Device responds with code HTTP 200 OK and **ModifyScheduleResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_schedules.modify_schedule

## 19.5 DELETE SCHEDULE

**Test Label:** Configure Schedules - Delete Schedule

**Test Case ID:** CONFIGURESCHEDULES-3

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Delete Schedule

**Test Purpose:** To verify that Client is able to delete schedule from Device using the **DeleteSchedule** operation.

**Pre-Requisite:**

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

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

1. Client invokes **DeleteSchedule** request message to delete schedule from the Device for specified schedule.
2. Device responds with code HTTP 200 OK and **DeleteScheduleResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_schedules.delete_schedule

# 20 Get Schedule State Test Cases

## 20.1 Feature Level Normative Reference:

**Validated Feature:** GetScheduleState

**Profile A Requirement:** Conditional

## 20.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a Schedule state using **GetscheduleState** operation OR using pull point mechanism.
2. Client is considered as supporting Get Schedule State if the following conditions are met:
   - Client is able to get Schedule state using **GetScheduleState** operation OR Client supports **tns1:Schedule/State/Active** notification if Device supports StateReporting AND
   - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature if Client supports **tns1:Schedule/State/Active** notification if Device supports StateReporting AND
3. Client is considered as NOT supporting Get Schedule State if ANY of the following is TRUE:
   - No valid responses for **GetScheduleState** request if detected if Device supports StateReporting OR
   - Client does not support **tns1:Schedule/State/Active** event AND Client unable to get Schedule state using **GetScheduleState** operation if Device supports StateReporting OR
   - Client does not support EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature when Client supports **tns1:Schedule/State/Active** notification if Device supports StateReporting.

## 20.3 GET SCHEDULE STATE

**Test Label:** Get Schedule Sate

**Test Case ID:** GETSCHEDULESTATE-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Schedule State

**Test Purpose:** To verify that credential state provided by Device is received by Client using the **GetScheduleState** operation.

**Pre-Requisite:**

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

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

1. Client invokes **GetScheduleState** request message to retrieve schedule state for specified schedule from the Device.
2. Device responds with code HTTP 200 OK and **GetScheduleStateResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_schedule_state.get_schedule_state

# 21 Reset Antipassback Violation Test Cases

## 21.1 Feature Level Normative Reference:

**Validated Feature:** ResetAntipassbackViolation

**Profile A Requirement:** Conditional

## 21.2 Expected Scenarios Under Test:

1. Client connects to Device to reset antipassback violation for a specified credential using **ResetAntipassbackViolation** operation.
2. Client is considered as supporting Reset Antipassback Violation if the following conditions are met:
   - Client is able to reset the antipassback violation of a credential using **ResetAntipassbackViolation** operation if Device supports ResetAntipassbackViolation AND
3. Client is considered as NOT supporting Reset Antipassback Violation if ANY of the following is TRUE:
   - No valid responses for **ResetAntipassbackViolation** request if Device supports ResetAntipassbackViolation.

## 21.3 RESET ANTIPASSBACK VIOLATIONS

**Test Label:** Reset Antipassback Violation

**Test Case ID:** RESETANTIPASSBACKVIOLATION-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Reset Antipassback Violation

**Test Purpose:** To verify that Client is able to reset antipassback violation using the **ResetAntipassbackViolation** operation.

**Pre-Requisite:**

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

- Device supports ResetAntipassbackViolation.

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

1. Client invokes **ResetAntipassbackViolation** request message to reset the antipassback violation of a credential on a Device.
2. Device responds with code HTTP 200 OK and **ResetAntipassbackViolationResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** reset_antipassback_violation.reset_antipassback_violation

# 22 Antipassback Violation Notifications Test Cases

## 22.1 Feature Level Normative Reference:

**Validated Feature:** AntipassbackViolationNotifications

**Profile A Requirement:** Conditional

## 22.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get antipassback violations notifications.

2. Client uses Pull Point event mechanism OR Basic Notification event mechanism (if Device supports Basic Notification event mechanism) to retrieve notification events from Device.

3. Client is considered as supporting Antipassback Violation Notifications if the following conditions are met:

   - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature AND

   - Client is able to retrieve tns1:Credential/State/ApbViolation notifications about antipassback violation if Device supports Credential service AND

4. Client is considered as NOT supporting Antipassback Violation Notifications if ANY of the following is TRUE:

   - Client does not support EventHandling_Pullpoint feature AND EventHandling_WS-BaseNotification feature OR

   - Client is not able to retrieve tns1:Credential/State/ApbViolation notifications about antipassback violation if Device supports Credential service.

# 23 Get Special Day Group List Test Cases

## 23.1 Feature Level Normative Reference:

**Validated Feature:** GetSpecialDayGroupList

**Profile A Requirement:** Conditional

## 23.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a complete list of Special Day Groups.
2. Client is considered as supporting Get Special Day Groups List if the following conditions are met:
   - Client is able to list available Special Day Groups using **GetSpecialDayGroupInfoList** operation OR **GetSpecialDayGroupList** operation.
3. Client is considered as NOT supporting Get Special Day Groups List if ANY of the following is TRUE:
   - No valid responses for **GetSpecialDayGroupInfoList** request OR **GetSpecialDayGroupList** request OR
   - **GetSpecialDayGroupInfoList** request contains **tsc:StartReference** element value that was not recieved in **GetSpecialDayGroupInfoList** response in **tsc:NextStartReference** element OR
   - **GetSpecialDayGroupList** request contains **tsc:StartReference** element value that was not recieved in **GetSpecialDayGroupList** response in **tsc:NextStartReference** element OR
   - Complete Special Day Groups list was not received.

## 23.3 LISTING OF SPECIAL DAY GROUPS

**Test Label:** Get Special Day Groups List - Listing of Special Day Groups

**Test Case ID:** GETSPECIALDAYGROUPLIST-1

**Profile A Normative Reference:** Optional

**Feature Under Test:** Get Special Day Groups List

**Test Purpose:** To verify that list of all special day groups items provided by Device is received by Client using the **GetSpecialDayGroupList** operation.

**Pre-Requisite:**

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

- Device supports SpecialDays.

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

1. Client invokes **GetSpecialDayGroupList** request message with skipped **tsc:StartReference** element to retrieve first part of the list of all special day groups configured on the Device.

2. Device responds with code HTTP 200 OK and **GetSpecialDayGroupListResponse** message.

3. If **GetSpecialDayGroupListResponse** message contains **tsc:NextStartReference** element Client invokes **GetSpecialDayGroupList** request message with **tsc:StartReference** element equal to value of **tsc:NextStartReference** element to retrieve next part of the list of all special day groups configured on the Device.

4. Client repeats the previous step while **GetSpecialDayGroupListResponse** message contains **tsc:NextStartReference** element.

**Test Result:**

**PASS -**

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

- First Client **GetSpecialDayGroupList** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tsc:GetSpecialDayGroupList** AND

  - [S2] It does not contain **tsc:StartReference** element AND

- If response on previous **GetSpecialDayGroupList** request contains **tsc:NextStartReference** element each next Client **GetSpecialDayGroupList** requests in Test Procedure fulfills the following requirements (else skip the checks):

  - [S3] **soapenv:Body** element has child element **tsc:GetSpecialDayGroupList** AND

  - [S4] It contains **tsc:StartReference** element equal to **tsc:NextStartReference** element from response on previous **GetSpecialDayGroupList** request AND

- Device responses on the each **GetSpecialDayGroupList** request in Test Procedure fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

  - [S6] **soapenv:Body** element has child element **tsc:GetSpecialDayGroupListResponse** AND

- The last in Test Procedure Device response on **GetSpecialDayGroupList** request fulfills the following requirements:

  - [S7] It does not contain **tsc:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_special_day_group_list.get_special_day_group_list

## 23.4 LISTING OF SPECIAL DAY GROUP INFO

**Test Label:** Get Special Day Groups List - Listing of Special Day Group Info

**Test Case ID:** GETSPECIALDAYGROUPLIST-2

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Special Day Group Info List

**Test Purpose:** To verify that list of all special day groups items provided by Device is received by Client using the **GetSpecialDayGroupInfoList** operation.

**Pre-Requisite:**

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

- Device supports SpecialDays.

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

1. Client invokes **GetSpecialDayGroupInfoList** request message with skipped **tsc:StartReference** element to retrieve first part of the list of all special day groups configured on the Device.

2. Device responds with code HTTP 200 OK and **GetSpecialDayGroupInfoListResponse** message.

3. If **GetSpecialDayGroupInfoListResponse** message contains **tsc:NextStartReference** element Client invokes **GetSpecialDayGroupInfoList** request message with **tsc:StartReference** element equal to value of **tsc:NextStartReference** element to retrieve next part of the list of all special day groups configured on the Device.

4. Client repeats the previous step while **GetSpecialDayGroupInfoListResponse** message contains **tsc:NextStartReference** element.

**Test Result:**

**PASS -**

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

- First Client **GetSpecialDayGroupInfoList** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tcr:GetSpecialDayGroupInfoList** AND

  - [S2] It does not contain **tsc:StartReference** element AND

- If response on previous **GetSpecialDayGroupInfoList** request contains **tsc:NextStartReference** element each next Client **GetSpecialDayGroupInfoList** requests in Test Procedure fulfills the following requirements (else skip the checks):

  - [S3] **soapenv:Body** element has child element **tsc:GetSpecialDayGroupInfoList** AND

  - [S4] It contains **tsc:StartReference** element equal to **tsc:NextStartReference** AND element from response on previous **GetSpecialDayGroupInfoList** request AND

- Device responses on the each **GetSpecialDayGroupInfoList** request in Test Procedure fulfills the following requirements:

  - [S5] It has HTTP 200 response code AND

  - [S6] **soapenv:Body** element has child element **tsc:GetSpecialDayGroupInfoListResponse** AND

- The last in Test Procedure Device response on **GetSpecialDayGroupInfoList** request fulfills the following requirements:

  - [S7] It does not contain **tsc:NextStartReference** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_special_day_group_list.get_special_day_group_info_list

# 24 Get Special Day Group Details Test Cases

## 24.1 Feature Level Normative Reference:

**Validated Feature:** GetSpecialDayGroupDetails

**Profile A Requirement:** Conditional

## 24.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve Special Day Groups details.

2. Client is considered as supporting Get Special Day Group Details if the following conditions are met:

   - Client is able to get special day groups details using **GetSpecialDayGroups** operation if Device supports SpecialDays.

3. Client is considered as NOT supporting Get Special Day Group Details if ANY of the following is TRUE:

   - No valid responses for **GetSpecialDayGroups** request with at least one spesial day group listed in it if Device supports SpecialDays.

## 24.3 GET SPECIAL DAY GROUPS

**Test Label:** Get Special Day Group Details - Get Special Day Groups

**Test Case ID:** GETSPECIALDAYGROUPDETAILS-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Get Special Day Groups

**Test Purpose:** To verify that Client is able to get special day groups detailes using the **GetSpecialDayGroups** operation.

**Pre-Requisite:**

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

- Device supports SpecialDays.

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

1. Client invokes **GetSpecialDayGroups** request message to get special day groups detailes from Device.

2. Device responds with code HTTP 200 OK and **GetSpecialDayGroups** message which contains at least one **Special Day Group** element.

**Test Result:**

**PASS -**

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

- Client **GetSpecialDayGroups** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tsc:GetSpecialDayGroups** AND

- Device response on the **GetSpecialDayGroups** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **tar:GetSpecialDayGroupsResponse** AND

  - [S4] **tsc:GetSpecialDayGroupsResponse** has at least one **tsc:SpecialDayGroup** element.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** get_special_day_group_details.get_special_day_groups

# 25 Configure Special Day Groups Test Cases

## 25.1 Feature Level Normative Reference:

**Validated Feature:** ConfigureSpecialDayGroups

**Profile A Requirement:** Conditional

## 25.2 Expected Scenarios Under Test:

1. Client creates special day group on a Device using **CreateSpecialDayGroup** operation.
2. Client modifys special day group on a Device using **ModifySpecialDayGroup** operation.
3. Client deletes special day group from a Device using **DeleteSpecialDayGroup** operation.
4. Client is considered as supporting Configure Special Day Groups if the following conditions are met:
   - Client is able to create special day group using **CreateSpecialDayGroup** operation if Device supports SpecialDays AND
   - Client is able to modify special day group using **ModifySpecialDayGroup** operation if Device supports SpecialDays AND
   - Client is able to delete special day group using **DeleteSpecialDayGroup** operation if Device supports SpecialDays.
5. Client is considered as NOT supporting Configure Special Day Groups if ANY of the following is TRUE:
   - No valid responses for **CreateSpecialDayGroup** request if Device supports SpecialDays OR
   - No valid responses for **ModifySpecialDayGroup** request if Device supports SpecialDays OR
   - No valid responses for **DeleteSpecialDayGroup** request if Device supports SpecialDays.

## 25.3 CREATE SPECIAL DAY GROUP

**Test Label:** Configure Special Day Groups - Create Special Day Group

**Test Case ID:** CONFIGURESPECIALDAYGROUPS-1

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Create Special Day Group

**Test Purpose:** To verify that Client is able to create a special day group on Device using the **CreateSpecialDayGroup** operation.

**Pre-Requisite:**

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

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

1. Client invokes **CreateSpecialDayGroup** request message to create special day group on Device.
2. Device responds with code HTTP 200 OK and **CreateSpecialDayGroupResponse** message.

**Test Result:**

**PASS -**

- Client **CreateSpecialDayGroup** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateSpecialDayGroup** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child **tsc:CreateSpecialDayGroup** element AND
  - [S2] **tsc:SpecialDayGroup/@token** attribute is empty (has empty string value) AND
- Device response on the **CreateSpecialDayGroup** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tsc:CreateSpecialDayGroupResponse**.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_special_day_group.create_special_day_group

## 25.4 MODIFY SPECIAL DAY GROUP

**Test Label:** Configure Special Day Groups - Modify Special Day Group

**Test Case ID:** CONFIGURESPECIALDAYGROUPS-2

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Modify Special Day Group

**Test Purpose:** To verify that Client is able to modify special day group on Device using the **ModifySpecialDayGroup** operation.

**Pre-Requisite:**

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

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

1. Client invokes **ModifySpecialDayGroup** request message to modify special day group on Device.
2. Device responds with code HTTP 200 OK and **ModifySpecialDayGroupResponse** message.

**Test Result:**

**PASS -**

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

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_special_day_group.modify_special_day_group

## 25.5 DELETE SPECIAL DAY GROUP

**Test Label:** Configure Special Day Groups - Delete Special Day Group

**Test Case ID:** CONFIGURESPECIALDAYGROUPS-3

**Profile A Normative Reference:** Conditional

**Feature Under Test:** Delete Special Day Group

**Test Purpose:** To verify that Client is able to delete a special day group from Device using the **DeleteSpecialDayGroup** operation.

**Pre-Requisite:**

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

- Device supports SpecialDays.

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

1. Client invokes **DeleteSpecialDayGroup** request message to delete special day group from the Device for specified special day group.

2. Device responds with code HTTP 200 OK and **DeleteSpecialDayGroupResponse** message.

**Test Result:**

**PASS -**

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

- Client **DeleteSpecialDayGroup** request in Test Procedure fulfills the following requirements:

  - [S1] **soapenv:Body** element has child element **tsc:DeleteSpecialDayGroup** AND

- Device response on the **DeleteSpecialDayGroup** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **tsc:DeleteSpecialDayGroupResponse**.

**FAIL -**

- The Client failed PASS criteria.

**Validated Feature List:** configure_special_day_group.delete_special_day_group

# 26 Special Days Notifications Test Cases

## 26.1 Feature Level Normative Reference:

**Validated Feature:** SpecialDaysNotifications

**Profile A Requirement:** Conditional

## 26.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using **CreatePullPointSubscription** operation OR **Subscribe** operation to get special days notifications.

2. Client uses Pull Point event mechanism OR Basic Notification event mechanism (if Device supports Basic Notification event mechanism) to retrieve notification events from Device.

3. Client is considered as supporting Special Days Notifications if the following conditions are met:

    - Client supports EventHandling_Pullpoint feature OR EventHandling_WS-BaseNotification feature AND

    - Client is able to retrieve tns1:Configuration/SpecialDays/Changed notifications about special days configuration change if Device supports SpecialDays AND

    - Client is able to retrieve tns1:Configuration/SpecialDays/Removed notifications about special days removing if Device supports SpecialDays AND

4. Client is considered as NOT supporting Special Days Notifications if ANY of the following is TRUE:

    - Client does not support EventHandling_Pullpoint feature AND EventHandling_WS-BaseNotification feature OR

    - Client is not able to retrieve tns1:Configuration/SpecialDays/Changed notifications about special days configuration change if Device supports SpecialDays OR

    - Client is not able to retrieve tns1:Configuration/SpecialDays/Removed notifications about special days removing if Device supports SpecialDays.