ONVIF®
Profile A Client Test Specification

Version 19.12

December 2019
Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

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

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

<table>
<thead>
<tr>
<th>Vers.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.12</td>
<td>Sep 18, 2019</td>
<td>The following was done according to #325: Supplementary Features and Test Cases sections was added.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supplementary Features and Test Cases sections was added.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 13, 2019</td>
<td>The following was done according to #325: EVENTHANDLING-3 METADATA STREAMING test was removed from Event Handling Feature and moved to Metadata Streaming Using Media2. Test case ID was changed to MEDIA2_METADATASTREAMING-1. Event Handling will use link to this test.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EVENTHANDLING-4 METADATA STREAMING USING MEDIA was added for Profile S Devices.</td>
</tr>
<tr>
<td>19.12</td>
<td>Sep 6, 2019</td>
<td>DEVICEDISCOVERYTYPEFILTER-1 DEVICE DISCOVERY TYPE FILTER was updated according to #323: Unnecessary step with check that ProbeMatch is sent to Client IP address was removed.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: HTTP Digest section and HTTP Digest Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: Capabilities section and Capabilities Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: Get Services section and Get Services Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: Event Handling section and Event Handling Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: Keep Alive for Pull Point Event Handling section and Keep Alive for Pull Point Event Handling Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>19.12</td>
<td>Aug 14, 2019</td>
<td>The following was done according to #341: Discovery section and Discovery Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
</tr>
<tr>
<td>Date</td>
<td>The following was done according to #341:</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Device Discovery Type Filter section and Device Discovery Type Filter Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>User Handling section and User Handling Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Network Configuration section and Network Configuration Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>System section and System Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Persistent Notification Storage Retrieval section and Persistent Notification Storage Retrieval Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Get Services with Capabilities section and Get Services with Capabilities Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Set Synchronization Point section and Set Synchronization Point Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Aug 14, 19</td>
<td>Unsubscribe section and Unsubscribe Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications.</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>
| 19.12 Aug 14, 2019 | The following was done according to #341:  
|              | Hostname Configuration section and Hostname Configuration Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications. |
| 19.12 Aug 14, 2019 | The following was done according to #341:  
|              | DNS Configuration section and DNS Configuration Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications. |
| 19.12 Aug 14, 2019 | The following was done according to #341:  
|              | Network Protocols Configuration section and Network Protocols Configuration Test Cases section was moved from ONVIF Core Client Test Specification to ONVIF Profile A Client Test Specifications. |
| 19.06 Jun 14, 2019 | The following was done according to #309:  
|              | 'Validated Feature' section for each feature updated to be synchronized with feature ID used in feature list.  
|              | 'Feature Under Test' section for each test case updated to be synchronized with sub-feature ID used in feature list.  
|              | 'Validated Feature List' test case section removed. |
| 18.06 Jun 21, 2018 | Reformatting document using new template |
| 18.06 Apr 05, 2018 | 'Required Number of Devices Summary' Annex added according to #241 |
| 18.06 Feb 16, 2018 | The following were updated in the scope of #241:  
|              | Feature Level Requirement (updated with new rules)  
|              | Each Feature Level Requirement (updated with Check Condition based on Device Features and Required Number of Devices) |
| 17.06 Jun 15, 2017 | Links in Normative references section were updated. |
| 16.12 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.  
|              | Antipassback Violations Notifications Test Cases added.  
|              | Special Days Notifications Test Cases added. |
| 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"  
|              | Get Access Profiles Details Test Cases added.  
<p>|              | Configure Access Profiles Test Cases added. |</p>
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.07</td>
<td>Mar 14, 2016</td>
<td>• <a href="http://www.onvif.org">www.onvif.org</a> was removed from Copyright section.</td>
</tr>
<tr>
<td>16.01</td>
<td>Dec 07, 2016</td>
<td>• General item (Test Overview) was added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Minor updates in formatting, typos and terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updates according review results (general changes): All test cases and use cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The following tests logic was updated to include logic for the case when all items were received in first GetXListResponse:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETCRENDENTIALLIST-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETCRENDENTIALLIST-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETSCHEDULELIST-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETSCHEDULELIST-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETACCESSPROFILELIST-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• GETACCESSPROFILELIST-2</td>
</tr>
<tr>
<td>15.10</td>
<td>Oct 13, 2016</td>
<td>• Initial version:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>General parts added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Credentials List Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Credentials Details Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Credential Configuration and State Notifications Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configure Credentials Test Cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Schedules List Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schedules Configuration Notifications Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Access Profiles List Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access Profiles Configuration Notifications Test Cases added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Get Credential Capabilities added</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** ...................................................................................................................... 16

   1.1 Scope ........................................................................................................................................ 16

1.2 Test Cases for Profile Mandatory Features .............................................................................. 17

   1.2.1 HTTP Digest ................................................................................................................ 17

   1.2.2 Capabilities .................................................................................................................. 17

   1.2.3 Get Services ................................................................................................................ 17

   1.2.4 Event Handling ............................................................................................................. 17

   1.2.5 Keep Alive for Pull Point Event Handling ..................................................................... 18

   1.2.6 Discovery ....................................................................................................................... 18

   1.2.7 Device Discovery Type Filter ...................................................................................... 18

   1.2.8 User Handling .............................................................................................................. 18

   1.2.9 Get Credential Capabilities .......................................................................................... 18

   1.2.10 Get Credential List ..................................................................................................... 18

   1.2.11 Get Credential Details ................................................................................................. 18

   1.2.12 Configure Credentials ................................................................................................. 18

   1.2.13 Credential Configuration and State Notifications ...................................................... 19

   1.2.14 Get Schedule List ....................................................................................................... 19

   1.2.15 Schedule Configuration Notifications ......................................................................... 19

   1.2.16 Get Access Profile List ............................................................................................... 19

   1.2.17 Access Profile Configuration Notifications .................................................................. 19

1.3 Test Cases for Profile Conditional Features ........................................................................... 19

   1.3.1 Network Configuration .................................................................................................. 19

   1.3.2 System ......................................................................................................................... 19

   1.3.3 IP Address Filtering ...................................................................................................... 19

   1.3.4 Persistent Notification Storage Retrieval ......................................................................... 20

   1.3.5 System Date and Time Configuration ............................................................................ 20

   1.3.6 Get Access Profile Details ............................................................................................ 20

   1.3.7 Configure Access Profiles ............................................................................................. 20

   1.3.8 Get Credential State ...................................................................................................... 20

   1.3.9 Change Credential State .................................................................................................. 20
1.3.10 Get Schedule Details ................................................................. 20
1.3.11 Configure Schedules ................................................................. 20
1.3.12 Get Schedule State ................................................................. 20
1.3.13 Reset Antipassback Violation ..................................................... 21
1.3.14 Antipassback Violation Notifications ........................................... 21
1.3.15 Get Special Day Group List ......................................................... 21
1.3.16 Get Special Day Group Details .................................................. 21
1.3.17 Configure Special Day Groups ................................................. 21
1.3.18 Special Days Notifications ....................................................... 21
1.4 Test Cases for Profile Optional Features ........................................ 21
  1.4.1 Get Services with Capabilities ...................................................... 21
  1.4.2 Set Synchronization Point .......................................................... 22
  1.4.3 Unsubscribe .............................................................................. 22
  1.4.4 Hostname Configuration ............................................................. 22
  1.4.5 DNS Configuration ..................................................................... 22
  1.4.6 Network Protocols Configuration .............................................. 22
1.5 Supplementary Features and Test Cases ......................................... 22

2 Normative references ............................................................................. 23

3 Terms and Definitions ............................................................................. 24
  3.1 Conventions ................................................................................ 24
  3.2 Definitions .................................................................................. 24
  3.3 Abbreviations ............................................................................. 25
  3.4 Namespaces ................................................................................. 26

4 Test Overview ......................................................................................... 28
  4.1 General ...................................................................................... 28
      4.1.1 Feature Level Requirement .................................................. 28
      4.1.2 Expected Scenarios Under Test ........................................... 28
      4.1.3 Test Cases .......................................................................... 29
  4.2 Test Setup ................................................................................. 29
  4.3 Prerequisites .............................................................................. 29

5 Test Cases for Profile Mandatory Features ......................................... 31
5.1 HTTP Digest Test Cases ................................................................. 31
  5.1.1 Feature Level Requirement: ..................................................... 31
  5.1.2 Expected Scenarios Under Test: ............................................. 31
  5.1.3 HTTP DIGEST ........................................................................ 32
5.2 Capabilities Test Cases ................................................................. 33
  5.2.1 Feature Level Requirement: ..................................................... 33
  5.2.2 Expected Scenarios Under Test: ............................................. 34
  5.2.3 GET SERVICES ................................................................. 34
  5.2.4 GET CAPABILITIES .................................................................. 35
5.3 Get Services Test Cases .............................................................. 36
  5.3.1 Feature Level Requirement: ..................................................... 36
  5.3.2 Expected Scenarios Under Test: ............................................. 37
5.4 Event Handling Test Cases .......................................................... 37
  5.4.1 Feature Level Requirement: ..................................................... 37
  5.4.2 Expected Scenarios Under Test: ............................................. 37
  5.4.3 PULLPOINT ................................................................. 38
  5.4.4 BASE NOTIFICATION .......................................................... 39
  5.4.5 METADATA STREAMING USING MEDIA .................................. 40
5.5 Keep Alive for Pull Point Event Handling Test Cases ...................... 43
  5.5.1 Feature Level Requirement: ..................................................... 43
  5.5.2 Expected Scenarios Under Test: ............................................. 44
  5.5.3 RENEW .................................................................................. 44
  5.5.4 PULL MESSAGES AS KEEP ALIVE ........................................ 46
5.6 Discovery Test Cases ................................................................. 47
  5.6.1 Feature Level Requirement: ..................................................... 47
  5.6.2 Expected Scenarios Under Test: ............................................. 48
  5.6.3 WS-DISCOVERY ............................................................... 48
5.7 Device Discovery Type Filter Test Cases ....................................... 49
  5.7.1 Feature Level Requirement: ..................................................... 49
  5.7.2 Expected Scenarios Under Test: ............................................. 50
  5.7.3 DEVICE DISCOVERY TYPE FILTER ..................................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.8</td>
<td>User Handling Test Cases</td>
</tr>
<tr>
<td>5.8.1</td>
<td>Feature Level Requirement:</td>
</tr>
<tr>
<td>5.8.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.8.3</td>
<td>CREATE USERS</td>
</tr>
<tr>
<td>5.8.4</td>
<td>GET USERS</td>
</tr>
<tr>
<td>5.8.5</td>
<td>SET USER</td>
</tr>
<tr>
<td>5.8.6</td>
<td>DELETE USERS</td>
</tr>
<tr>
<td>5.9</td>
<td>Get Credential Capabilities Test Cases</td>
</tr>
<tr>
<td>5.9.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.9.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.9.3</td>
<td>GET SERVICE CAPABILITIES</td>
</tr>
<tr>
<td>5.10</td>
<td>Get Credential List Test Cases</td>
</tr>
<tr>
<td>5.10.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.10.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.10.3</td>
<td>LISTING OF CREDENTIALS</td>
</tr>
<tr>
<td>5.10.4</td>
<td>LISTING OF CREDENTIAL INFO</td>
</tr>
<tr>
<td>5.11</td>
<td>Get Credential Details Test Cases</td>
</tr>
<tr>
<td>5.11.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.11.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.11.3</td>
<td>GET CREDENTIALS</td>
</tr>
<tr>
<td>5.12</td>
<td>Configure Credentials Test Cases</td>
</tr>
<tr>
<td>5.12.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.12.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.12.3</td>
<td>GET SUPPORTED FORMAT TYPES</td>
</tr>
<tr>
<td>5.12.4</td>
<td>CREATE CREDENTIAL</td>
</tr>
<tr>
<td>5.12.5</td>
<td>MODIFY CREDENTIAL</td>
</tr>
<tr>
<td>5.12.6</td>
<td>DELETE CREDENTIAL</td>
</tr>
<tr>
<td>5.13</td>
<td>Credential Configuration and State Notifications Test Cases</td>
</tr>
<tr>
<td>5.13.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.13.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.14</td>
<td>Get Schedule List Test Cases</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>5.14.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.14.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.14.3</td>
<td>LISTING OF SCHEDULES</td>
</tr>
<tr>
<td>5.14.4</td>
<td>LISTING OF SCHEDULE INFO</td>
</tr>
<tr>
<td>5.15</td>
<td>Schedule Configuration Notifications Test Cases</td>
</tr>
<tr>
<td>5.15.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.15.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.16</td>
<td>Get Access Profile List Test Cases</td>
</tr>
<tr>
<td>5.16.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.16.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>5.16.3</td>
<td>LISTING OF ACCESS PROFILES</td>
</tr>
<tr>
<td>5.16.4</td>
<td>LISTING OF ACCESSPROFILE INFO</td>
</tr>
<tr>
<td>5.17</td>
<td>Access Profile Configuration Notifications Test Cases</td>
</tr>
<tr>
<td>5.17.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>5.17.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6</td>
<td>Test Cases for Profile Conditional Features</td>
</tr>
<tr>
<td>6.1</td>
<td>Network Configuration Test Cases</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Feature Level Requirement:</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.1.3</td>
<td>GET NETWORK INTERFACES</td>
</tr>
<tr>
<td>6.1.4</td>
<td>SET NETWORK INTERFACES</td>
</tr>
<tr>
<td>6.1.5</td>
<td>GET NETWORK DEFAULT GATEWAY</td>
</tr>
<tr>
<td>6.1.6</td>
<td>SET NETWORK DEFAULT GATEWAY</td>
</tr>
<tr>
<td>6.2</td>
<td>System Test Cases</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Feature Level Requirement:</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.2.3</td>
<td>GET DEVICE INFORMATION</td>
</tr>
<tr>
<td>6.3</td>
<td>IP Address Filtering Test Cases</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Feature Level Requirement:</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.3.3</td>
<td>GET IP ADDRESS FILTER</td>
</tr>
</tbody>
</table>
6.3.4 SET IPv4 ADDRESS FILTER .......................................................... 94
6.3.5 SET IPv6 ADDRESS FILTER .......................................................... 95
6.3.6 ADD IPv4 ADDRESS FILTER .......................................................... 96
6.3.7 ADD IPv6 ADDRESS FILTER .......................................................... 98
6.3.8 REMOVE IPv4 ADDRESS FILTER ................................................. 99
6.3.9 REMOVE IPv6 ADDRESS FILTER ................................................. 100

6.4 Persistent Notification Storage Retrieval Test Cases ....................... 101
6.4.1 Feature Level Requirement: ...................................................... 101
6.4.2 Expected Scenarios Under Test: ............................................... 102
6.4.3 SEEK ......................................................................................... 102

6.5 System Date and Time Configuration Test Cases ................................ 104
6.5.1 Feature Level Requirement: ...................................................... 104
6.5.2 Expected Scenarios Under Test: ............................................... 104
6.5.3 GET SYSTEM DATE AND TIME ............................................... 105
6.5.4 SET SYSTEM DATE AND TIME ............................................... 106

6.6 Get Access Profile Details Test Cases .............................................. 107
6.6.1 Feature Level Normative Reference: ........................................... 107
6.6.2 Expected Scenarios Under Test: ............................................... 107
6.6.3 GET ACCESS PROFILES ......................................................... 108

6.7 Configure Access Profiles Test Cases .............................................. 109
6.7.1 Feature Level Normative Reference: ........................................... 109
6.7.2 Expected Scenarios Under Test: ............................................... 109
6.7.3 CREATE ACCESS PROFILE ..................................................... 110
6.7.4 MODIFY ACCESS PROFILE .................................................. 111
6.7.5 DELETE ACCESS PROFILE .................................................... 112

6.8 Get Credential State Test Cases .................................................... 113
6.8.1 Feature Level Normative Reference: ........................................... 113
6.8.2 Expected Scenarios Under Test: ............................................... 113
6.8.3 GET CREDENTIAL STATE ....................................................... 113

6.9 Change Credential State Test Cases ............................................... 114
6.9.1 Feature Level Normative Reference: ........................................... 114
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.9.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.9.3</td>
<td>ENABLE CREDENTIAL</td>
</tr>
<tr>
<td>6.9.4</td>
<td>DISABLE CREDENTIAL</td>
</tr>
<tr>
<td>6.10</td>
<td>Get Schedule Details Test Cases</td>
</tr>
<tr>
<td>6.10.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.10.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.10.3</td>
<td>GET SCHEDULES</td>
</tr>
<tr>
<td>6.11</td>
<td>Configure Schedules Test Cases</td>
</tr>
<tr>
<td>6.11.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.11.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.11.3</td>
<td>CREATE SCHEDULE</td>
</tr>
<tr>
<td>6.11.4</td>
<td>MODIFY SCHEDULE</td>
</tr>
<tr>
<td>6.11.5</td>
<td>DELETE SCHEDULE</td>
</tr>
<tr>
<td>6.12</td>
<td>Get Schedule State Test Cases</td>
</tr>
<tr>
<td>6.12.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.12.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.12.3</td>
<td>GET SCHEDULE STATE</td>
</tr>
<tr>
<td>6.13</td>
<td>Reset Antipassback Violation Test Cases</td>
</tr>
<tr>
<td>6.13.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.13.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.13.3</td>
<td>RESET ANTIPASSBACK VIOLATIONS</td>
</tr>
<tr>
<td>6.14</td>
<td>Antipassback Violation Notifications Test Cases</td>
</tr>
<tr>
<td>6.14.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.14.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.15</td>
<td>Get Special Day Group List Test Cases</td>
</tr>
<tr>
<td>6.15.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
<tr>
<td>6.15.2</td>
<td>Expected Scenarios Under Test:</td>
</tr>
<tr>
<td>6.15.3</td>
<td>LISTING OF SPECIAL DAY GROUPS</td>
</tr>
<tr>
<td>6.15.4</td>
<td>LISTING OF SPECIAL DAY GROUP INFO</td>
</tr>
<tr>
<td>6.16</td>
<td>Get Special Day Group Details Test Cases</td>
</tr>
<tr>
<td>6.16.1</td>
<td>Feature Level Normative Reference:</td>
</tr>
</tbody>
</table>
7.5.3 GET DNS ................................................................. 148
7.5.4 SET DNS ................................................................. 149

7.6 Network Protocols Configuration Test Cases ........................................ 151
7.6.1 Feature Level Requirement: ................................................................. 151
7.6.2 Expected Scenarios Under Test: ............................................................. 151
7.6.3 GET NETWORK PROTOCOLS ......................................................... 151
7.6.4 SET NETWORK PROTOCOLS ......................................................... 153

8 Supplementary Features and Test Cases ...................................................... 155
8.1 METADATA STREAMING USING MEDIA2 ........................................... 155

A Test for Appendix A .................................................................................. 158
A.1 Required Number of Devices Summary .................................................... 158
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 Test Cases for Profile Mandatory Features

This section defines test cases which are mandatory for Profile A Client conformance.

1.2.1 HTTP Digest

HTTP Digest section defines security mechanism for HTTP Digest Authentication.

1.2.2 Capabilities

Capabilities section specifies Client ability to retrieve available services and advanced functionalities which are offered by a Device.

1.2.3 Get Services

Get Services section specifies Client ability to retrieve list of services with using GetServices operation.

1.2.4 Event Handling

Event Handling section defines Client ability to initiate and receive notifications (events) from a Device.

The event handling test cases cover the following mandatory interfaces:

- Pull Point Notification Interface
  - This test specification provides test cases to verify the implementation of the PullPoint Interface of a Client.

- Basic Notification Interface
  - This test specification provides test cases to verify the implementation of the Basic Notification Interface of a Client.

- Metadata Streaming Interface
  - This test specification provides test cases to verify the implementation of the Metadata Streaming Interface of a Client using Media Service and using Media2 Service.
1.2.5 Keep Alive for Pull Point Event Handling

Keep Alive for Pull Point Event Handling section specifies Client ability to use keep alive for Pull Point Event Handling using PullMessages or Renew approach.

1.2.6 Discovery

Discovery section defines Client ability to locate services on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IP multicast address 239.255.255.250 and TCP and UDP port 3702 and SOAP-over-UDP standard for communication between nodes.

1.2.7 Device Discovery Type Filter

Device Discovery Type Filter Test Cases section defines Client ability to locate services, which are support Device Discovery Type on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter is equal to tds:Device or with skipped Types filter.

1.2.8 User Handling

User Handling section defines Client ability to manage users on Device.

1.2.9 Get Credential Capabilities

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

1.2.10 Get Credential List

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

1.2.11 Get Credential Details

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

1.2.12 Configure Credentials

Configure Credentials section specifies Client ability configure Credentials on Device.
1.2.13 Credential Configuration and State Notifications

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

1.2.14 Get Schedule List

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

1.2.15 Schedule Configuration Notifications

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

1.2.16 Get Access Profile List

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

1.2.17 Access Profile Configuration Notifications

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

1.3 Test Cases for Profile Conditional Features

This section defines test cases which are mandatory for Profile A Client conformance.

1.3.1 Network Configuration

Network Configuration section defines Client ability to obtain and configure of network settings on Device.

1.3.2 System

System section defines Client ability to obtain Device information and configure of system settings on Device.

1.3.3 IP Address Filtering

IP Address Filtering section defines Client ability to manage IP address filters on Device.
1.3.4 Persistent Notification Storage Retrieval

Persistent Notification Storage Retrieval section defines Client ability to seek stored events in Device.

1.3.5 System Date and Time Configuration

System Date and Time Configuration section defines Client ability to configure Device system date and time using GetSystemDateAndTime and SetSystemDateAndTime operations.

1.3.6 Get Access Profile Details

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

1.3.7 Configure Access Profiles

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

1.3.8 Get Credential State

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

1.3.9 Change Credential State

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

1.3.10 Get Schedule Details

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

1.3.11 Configure Schedules

Configure Schedules section specifies Client ability configure Schedules on Device.

1.3.12 Get Schedule State

Get Schedule State section specifies Client ability to get schedule state.
1.3.13 Reset Antipassback Violation

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

1.3.14 Antipassback Violation Notifications

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

1.3.15 Get Special Day Group List

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

1.3.16 Get Special Day Group Details

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

1.3.17 Configure Special Day Groups

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

1.3.18 Special Days Notifications

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

1.4 Test Cases for Profile Optional Features

This section defines test cases which are optional for Profile A Client conformance.

1.4.1 Get Services with Capabilities

Get Services with Capabilities section specifies Client ability to retrieve capabilities of services with using GetServices operation.
1.4.2 Set Synchronization Point

Set Synchronization Point section defines Client ability to synchronize its properties with the properties of the device using SetSynchronizationPoint operation.

1.4.3 Unsubscribe

Unsubscribe section defines Client ability to terminate subscription using Unsubscribe operation.

1.4.4 Hostname Configuration

Hostname Configuration section defines Client ability to obtain and configure of hostname settings on Device.

1.4.5 DNS Configuration

DNS Configuration section defines Client ability to obtain and configure of DNS settings on Device.

1.4.6 Network Protocols Configuration

Network Protocols Configuration section defines Client ability to obtain and configure of network protocols settings on Device.

1.5 Supplementary Features and Test Cases

This section defines supplementary features and test cases which are not the part of profile, but Profile A Features results depends on them.
2 Normative references

- ONVIF Conformance Process Specification:
  https://www.onvif.org/profiles/conformance/

- ONVIF Profile Policy:
  https://www.onvif.org/profiles/

- ONVIF Network Interface Specifications:
  https://www.onvif.org/profiles/specifications/

- ISO/IEC Directives, Part 2, Annex H:
  www.iso.org/directives

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

- 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/]

- W3C Web Services Addressing 1.0 – Core:
  http://www.w3.org/TR/ws-addr-core/

- ONVIF Profile A Specification:
  https://www.onvif.org/profiles/profile-a/
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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>An address refers to a URI.</td>
</tr>
<tr>
<td>Profile</td>
<td>See ONVIF Profile Policy.</td>
</tr>
<tr>
<td>ONVIF Device</td>
<td>Computer appliance or software program that exposes one or multiple ONVIF Web Services.</td>
</tr>
<tr>
<td>ONVIF Client</td>
<td>Computer appliance or software program that uses ONVIF Web Services.</td>
</tr>
<tr>
<td>Conversation</td>
<td>A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.</td>
</tr>
<tr>
<td>Network</td>
<td>A network is an interconnected group of devices communicating using the Internet protocol.</td>
</tr>
<tr>
<td>Network Trace Capture file</td>
<td>Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.</td>
</tr>
<tr>
<td>SOAP</td>
<td>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.</td>
</tr>
<tr>
<td>Client Test Tool</td>
<td>ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.</td>
</tr>
<tr>
<td>Valid Device Response</td>
<td>Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.</td>
</tr>
<tr>
<td>Profile A</td>
<td>The Profile A Specification.</td>
</tr>
<tr>
<td>Access Policy</td>
<td>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.</td>
</tr>
<tr>
<td>Access Profile</td>
<td>A collection of access policies, used to define role based access.</td>
</tr>
<tr>
<td>Access Point</td>
<td>A logical composition of a physical door and ID point(s) controlling access in one direction.</td>
</tr>
<tr>
<td>Credential</td>
<td>A physical/tangible object, a piece of knowledge, or a facet of a person's physical being, that enables an individual access</td>
</tr>
</tbody>
</table>
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.

3.3 Abbreviations
This section describes abbreviations used in this document.

HTTP
Hyper Text Transport Protocol.
HTTPS  Hyper Text Transport Protocol over Secure Socket Layer.
IP  Internet Protocol.
TCP  Transport Control Protocol.
UDP  User Datagram Protocol.
URI  Uniform Resource Identifier.
WSDL  Web Services Description Language.
XML  eXtensible Markup Language.
PACS  Physical Access Control System.

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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>soapenv</td>
<td><a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></td>
<td>Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]</td>
</tr>
<tr>
<td>xsi</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
<td>XML schema instance namespace</td>
</tr>
<tr>
<td>tns1</td>
<td><a href="http://www.onvif.org/ver10/topics">http://www.onvif.org/ver10/topics</a></td>
<td>The namespace for the ONVIF topic namespace</td>
</tr>
<tr>
<td>tt</td>
<td><a href="http://www.onvif.org/ver10/schema">http://www.onvif.org/ver10/schema</a></td>
<td>ONVIF XML schema descriptions</td>
</tr>
<tr>
<td>tds</td>
<td><a href="http://www.onvif.org/ver10/device/wsd">http://www.onvif.org/ver10/device/wsd</a></td>
<td>The namespace for the WSDL device service</td>
</tr>
<tr>
<td>tev</td>
<td><a href="http://www.onvif.org/ver10/events/wsd">http://www.onvif.org/ver10/events/wsd</a></td>
<td>The namespace for the WSDL event service</td>
</tr>
<tr>
<td>ter</td>
<td><a href="http://www.onvif.org/ver10/error">http://www.onvif.org/ver10/error</a></td>
<td>The namespace for ONVIF defined faults</td>
</tr>
<tr>
<td>wsa</td>
<td><a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a></td>
<td>Device addressing namespace as defined by [WS-Addressing].</td>
</tr>
<tr>
<td>tac</td>
<td><a href="http://www.onvif.org/ver10/accesscontrol/wsd">http://www.onvif.org/ver10/accesscontrol/wsd</a></td>
<td>The namespace for the WSDL access control service</td>
</tr>
<tr>
<td>tdc</td>
<td><a href="http://www.onvif.org/ver10/doorcontrol/wsd">http://www.onvif.org/ver10/doorcontrol/wsd</a></td>
<td>The namespace for the WSDL door control service</td>
</tr>
<tr>
<td>Prefix</td>
<td>Namespace URI</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>tar</td>
<td><a href="http://www.onvif.org/ver10/accessrules/wsdl">http://www.onvif.org/ver10/accessrules/wsdl</a></td>
<td>The namespace for the WSDL access rules service</td>
</tr>
<tr>
<td>tcr</td>
<td><a href="http://www.onvif.org/ver10/credential/wsdl">http://www.onvif.org/ver10/credential/wsdl</a></td>
<td>The namespace for the WSDL credential service</td>
</tr>
<tr>
<td>tsc</td>
<td><a href="http://www.onvif.org/ver10/schedule/wsdl">http://www.onvif.org/ver10/schedule/wsdl</a></td>
<td>The namespace for the WSDL schedule service</td>
</tr>
</tbody>
</table>
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 Requirement
- Expected Scenarios Under Test
- List of Test Cases

4.1.1 Feature Level Requirement

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

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

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

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

4.1.2 Expected Scenarios Under Test

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

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

Each Test Case contains the following parts:

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

4.2 Test Setup

Collect Network traces files required by the test cases.

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

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

For compatibility with the Profile 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 Test Cases for Profile Mandatory Features

5.1 HTTP Digest Test Cases

5.1.1 Feature Level Requirement:

Validated Feature: HTTP Digest authentication (HTTPDigest)

Check Condition based on Device Features: Digest

Required Number of Devices: 3

Profile A Requirement: Mandatory
Profile C Requirement: Mandatory
Profile D Requirement: Mandatory
Profile G Requirement: Mandatory
Profile Q Requirement: Mandatory
Profile S Requirement: Mandatory
Profile T Requirement: Mandatory
Profile M Requirement: Mandatory

5.1.2 Expected Scenarios Under Test:

1. Client invokes a specific command which is under testing without any user credentials (no UsernameToken, no HTTP Digest authentication header).

2. Device returns HTTP 401 Unauthorized error along with WWW-Authentication: Digest header.

3. Client re-sends request with HTTP Digest Authentication header corresponding to header provided in device response.

4. Device sends a valid response to this request.

5. Client is considered as supporting HTTP Digest if the following conditions are met:
   • Device returns a valid response to specific request with HTTP Digest authentication header.
6. Client is considered as NOT supporting HTTP Digest if the following is TRUE:
   • All HTTP Digest attempts detected are failed.

5.1.3 HTTP DIGEST

Test Label: Security - HTTP Digest Authentication.

Test Case ID: HTTPDIGEST-1

Feature Under Test: HTTP Digest (HTTPDigest_HTTPDigestAuthentication)

Profile S Normative Reference: Mandatory
Profile G Normative Reference: Mandatory
Profile C Normative Reference: Mandatory
Profile Q Normative Reference: Mandatory
Profile A Normative Reference: Mandatory
Profile T Normative Reference: Mandatory
Profile D Normative Reference: Mandatory
Profile M Normative Reference: Mandatory

Test Purpose: To verify that the Client supports the HTTP Digest Authentication for HTTP level security.

Pre-Requisite:
   • The Network Trace Capture files contains at least one Conversation between Client and Device with HTTP Digest Authentication present.

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

1. Client sends a request that requires authentication (e.g. GetUsers) to the Device without any authentication.
2. Device rejects the request with HTTP error code 401 AND an HTTP Digest challenge.
3. Client sends a valid request with HTTP Digest Authentication.
4. Device accepts the correct request with response code HTTP 200 OK.

Test Result:

PASS -
• [S1] Client request contains (HTTP GET method OR HTTP POST method) without any authentication AND

• Client HTTP GET request has a proper hierarchy (refer to [RFC 1945]) AND
  • [S2] Device response contains "HTTP/* 401 Unauthorized" AND
  • [S3] Device response contains "realm=" element AND
  • [S4] Device response contains "nonce=" element AND
  • [S5] Client request contains (HTTP GET method OR HTTP POST method) with "Authorization: Digest username=" element AND

• Client HTTP GET request with HTTP Authentication has a proper hierarchy (refer to [RFC 1945]) AND
  • [S6] Client request contains "realm=" element with value from Device response AND
  • [S7] Client request contains "nonce=" element with value from Device response AND
  • [S8] Client request contains "uri=" element AND
  • [S9] Device response contains "HTTP/* 200 OK".

FAIL -
  • The Client failed PASS criteria.

5.2 Capabilities Test Cases

5.2.1 Feature Level Requirement:

Validated Feature: Capabilities (Capabilities)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile C Requirement: Mandatory

Profile G Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile S Requirement: Mandatory
5.2.2 Expected Scenarios Under Test:

1. Client invokes a specific Capabilities command which is under testing.

2. Client is considered as supporting Capabilities if the following conditions are met:
   - Device returns a valid response to GetServices request OR
   - Device returns a valid response to GetCapabilities request.

3. Client is considered as NOT supporting Capabilities if the following is TRUE:
   - No Valid Device Response to GetServices request AND
   - No Valid Device Response to GetCapabilities request.

5.2.3 GET SERVICES

Test Label: Capabilities - Determine the available Services

Test Case ID: CAPABILITIES-1

Feature Under Test: Get Services (Capabilities_GetServicesRequest)

Profile S Normative Reference: Mandatory
Profile G Normative Reference: Mandatory
Profile C Normative Reference: Mandatory
Profile Q Normative Reference: Mandatory
Profile A Normative Reference: Mandatory
Profile T Normative Reference: Mandatory
Profile M Normative Reference: Mandatory

Test Purpose: To verify that Device Capabilities is received using GetServices request.

Pre-Requisite:
- The Network Trace Capture files contains at least one Conversation between Client and Device with GetServices command present.

Test Procedure (expected to be reflected in network trace file):
1. Client invokes GetServices request message to retrieve all services of the Device.

2. Verify that GetServicesResponse message from the Device contains code HTTP 200 OK without SOAP Fault.

Test Result:

PASS -

- Client GetServices request messages are valid according to XML Schemas listed in Namespaces AND
- Client GetServices request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetServices>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND

FAIL -

- The Client failed PASS criteria.

5.2.4 GET CAPABILITIES

Test Label: Capabilities - Get Device Capabilities

Test Case ID: CAPABILITIES-2

Feature Under Test: Get Capabilities (Capabilities_GetCapabilities)

Profile S Normative Reference: Mandatory

Profile G Normative Reference: Optional

Profile C Normative Reference: Optional

Profile T Normative Reference: None

Test Purpose: To verify that Device Capabilities is received using GetCapabilities request.

Pre-Requisite:

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

Test Procedure (expected to be reflected in network trace file):
1. Client invokes GetCapabilities request message to retrieve Device Capabilities of the Device.

2. Verify that GetCapabilitiesResponse response message from the Device contains code HTTP 200 OK without SOAP Fault.

Test Result:

PASS -

• Client GetCapabilities request messages are valid according to XML Schemas listed in Namespaces AND

• Client GetCapabilities request in Test Procedure fulfills the following requirements:
  • [S1] Client request contains "<GetCapabilities>" tag after the "<Body>" tag AND
  • [S2] Device response contains "HTTP/* 200 OK" AND
  • [S3] Device response contains "<GetCapabilitiesResponse>" tag.

FAIL -

• The Client failed PASS criteria.

5.3 Get Services Test Cases

5.3.1 Feature Level Requirement:

Validated Feature: Get Services (GetServices)

Check Condition based on Device Features: GetServices is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile D Requirement: Mandatory

Profile C Requirement: Mandatory

Profile G Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory
5.3.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a services using **GetServices** command.

2. Client is considered as supporting Get Services if the following conditions are met:
   - Client supports Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).

3. Client is considered as NOT supporting Get Services if ANY of the following is TRUE:
   - Client does not support Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).

5.4 Event Handling Test Cases

5.4.1 Feature Level Requirement:

**Validated Feature:** Event Handling (EventHandling)

**Check Condition based on Device Features:** None

**Required Number of Devices:** 3

**Profile S Requirement:** Conditional

**Profile G Requirement:** Conditional

**Profile Q Requirement:** Conditional

**Profile A Requirement:** Mandatory

**Profile C Requirement:** Mandatory

**Profile T Requirement:** Mandatory

**Profile D Requirement:** Mandatory

5.4.2 Expected Scenarios Under Test:

1. Client connects to Device to initiate Event Handling.

2. Client is considered as supporting Event Handling if the following conditions are met:
   - Client is able to handle the Pull Point Event mechanism OR
• Client is able to handle the Base Notification Event mechanism OR

• Client is able to handle the Metadata Streaming by supporting EventHandling_MetadataStreamingUsingMedia feature (please see EVENTHANDLING-4 METADATA STREAMING USING MEDIA section) OR Media2_MetadataStreaming_MetadataStreamingUsingMedia2 feature (please see MEDIA2_METADATASTREAMING-1 METADATA STREAMING USING MEDIA2 section).

3. Client is considered as NOT supporting Event Handling if the following is TRUE:

   • All Pull Point attempts detected have failed AND
   • All Base Notification attempts detected have failed AND
   • All Metadata Streaming attempts detected have failed.

5.4.3 PULLPOINT

Test Label: Event Handling - Pull Point

Test Case ID: EVENTHANDLING-1

Feature Under Test: Pull Point (EventHandling_PullPoint)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Governed by business rule #3

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Test Purpose: To verify that the Client is able to retrieve events using Pull Point.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with Pull Point event type.

Test Procedure (expected to be reflected in network trace file):
1. Client invokes CreatePullPointSubscription message.

2. Device responds with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.


Test Result:

PASS -

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

- Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<CreatePullPointSubscription>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<CreatePullPointSubscriptionResponse>" tag AND

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

- Client PullMessages request in Test Procedure fulfills the following requirements:
  - [S4] Client request contains "<PullMessages>" tag after the "<Body>" tag AND
  - [S7] Device response contains "HTTP/* 200 OK" AND

FAIL -

- The Client failed PASS criteria.

5.4.4 BASE NOTIFICATION

Test Label: Event Handling - Basic Notification

Test Case ID: EVENTHANDLING-2

Feature Under Test: Base Notification (EventHandling_WSBaseNotification)
Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Governed by business rule #3

Profile Q Normative Reference: None

Profile A Normative Reference: None

Profile T Normative Reference: None

Test Purpose: To verify that the Client is able to retrieve events using WS-Base Notification.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with Basic Notification event type.

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

1. Client invokes Subscribe message with ConsumerReference element.

2. Device responds with code HTTP 200 OK and SubscribeResponse message.

Test Result:

PASS -

• Client Subscribe request messages are valid according to XML Schemas listed in Namespaces AND

• Client Subscribe request in Test Procedure fulfills the following requirements:

  • [S1] Client request contains "<Subscribe>" tag after the "<Body>" tag AND

  • [S4] Device response contains "HTTP/* 200 OK" AND


FAIL -

• The Client failed PASS criteria.

5.4.5 METADATA STREAMING USING MEDIA

Test Label: Event Handling - Metadata Streaming Using Media Streaming

Test Case ID: EVENTHANDLING-4
Feature Under Test: Metadata Streaming (EventHandling_MetadataStreamingUsingMedia)

Profile S Normative Reference: Conditional

Profile G Normative Reference: None

Profile C Normative Reference: None

Profile Q Normative Reference: None

Profile A Normative Reference: None

Profile T Normative Reference: None

Test Purpose: To verify that the Client is able to retrieve the Metadata Streaming using Media Service.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with Metadata Streaming event type using Media Service.

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

1. Client invokes GetStreamUri request message for Media service for media profile that contains Video Source Configuration and Metadata Configuration. GetStreamUri request is set for RTP-Unicast/UDP OR RTP-Multicast/UDP OR RTP/RTSP/TCP OR RTP-Unicast/RTSP/HTTP/TCP transport.

2. Device responds with code HTTP 200 OK and GetStreamUriResponse message.

3. Client invokes RTSP DESCRIBE request to retrieve media stream description.

4. Device responds with code RTSP 200 OK and SDP information with Media Type: "application" and with encoding name "vnd.onvif.metadata" or "vnd.onvif.metadata.gzip" or "vnd.onvif.metadata.exi.onvif" or "vnd.onvif.metadata.exi.ext".

5. Client invokes RTSP SETUP request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for metadata streaming.

6. Device responds with code RTSP 200 OK.

7. Client invokes RTSP PLAY request without "onvif-replay" Require header to start media stream.

8. Device responds with code RTSP 200 OK.

9. Client invokes RTSP TEARDOWN request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client RTSP DESCRIBE request in Test Procedure

- Device response on the RTSP DESCRIBE request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "application" (m=application) with sessions attribute "rtpmap" with encoding name "vnd.onvif.metadata" OR "vnd.onvif.metadata.gzip" OR "vnd.onvif.metadata.exi.onvif" OR "vnd.onvif.metadata.exi.ext" (see ONVIF Streaming Spec) AND

- There is Client RTSP SETUP request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client RTSP DESCRIBE request AND
  - [S4] It invoked after the Client RTSP DESCRIBE request AND
  - [S5] RTSP address that was used to send RTSP SETUP is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain Require request header field with value is equal to "onvif-replay" AND

- Device response on the RTSP SETUP request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND

- There is a Device response on the GetStreamUri request invoked for Media Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client RTSP DESCRIBE request AND
  - [S10] It received before the Client RTSP DESCRIBE request AND
  - [S11] It contains trt:MediaUri|tt:Uri element which value is equal to RTSP address that was used to send the RTSP DESCRIBE request AND
• There is Client RTSP PLAY request in Test Procedure fulfills the following requirements:
  • [S12] It invoked for the same Device as for the Client RTSP SETUP request AND
  • [S13] It invoked after the Client RTSP SETUP request AND
  • [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  • [S15] It does not contain Require request header field with value is equal to "onvif-replay" AND

• Device response on the RTSP PLAY request fulfills the following requirements:
  • [S16] It has RTSP 200 response code AND

• There is Client RTSP TEARDOWN request in Test Procedure fulfills the following requirements:
  • [S17] It invoked for the same Device as for the Client RTSP SETUP request AND
  • [S18] It invoked after the Client RTSP PLAY request AND
  • [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

  • If there is Device response on the RTSP TEARDOWN request then it fulfills the following requirements:
    • [S20] It has RTSP 200 response code.

FAIL -
  • The Client failed PASS criteria.

5.5 Keep Alive for Pull Point Event Handling Test Cases

5.5.1 Feature Level Requirement:

Validated Feature: Keep Alive for Pull Point Event Handling (KeepAliveForPullPointEventHandling)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile A Requirement: Mandatory
Profile C Requirement: Mandatory
Profile S Requirement: Conditional
Profile Q Requirement: Optional
Profile G Requirement: Conditional
Profile T Requirement: Optional

5.5.2 Expected Scenarios Under Test:

1. Client connects to Device to initiate Pull Point Event Handling.

2. Client is considered as supporting Keep Alive for Pull Point Event Handling if the following conditions are met:
   - Client supports EventHandling_Pullpoint feature AND
   - Client is able to renew pull point subscription using Renew operation OR PullMessages operation mechanism.

3. Client is considered as NOT supporting Keep Alive for Pull Point Event Handling if the following is TRUE:
   - No valid responses for Renew request AND for CreatePullPointSubscription request in the case if PullMessages used for keep alive OR
   - No valid responses for Renew request if detected OR
   - No valid responses for CreatePullPointSubscription request in the case if PullMessages used for keep alive if detected OR
   - Renew request was invoked to address which was not specified in tev:SubscriptionReference\wsa:Address element of corresponding CreatePullPointSubscriptionResponse message.

5.5.3 RENEW

Test Label: Advanced Pull Point Event Handling - Renew

Test Case ID: KEEPALIVEFORPULLPOINTEVENTHANDLING-1
Feature Under Test: Renew (KeepAliveForPullPointEventHandling_Renew)

Profile A Normative Reference: Mandatory

Profile C Normative Reference: Mandatory

Profile S Normative Reference: Conditional

Profile Q Normative Reference: Optional

Profile G Normative Reference: Conditional

Profile T Normative Reference: Optional

Test Purpose: To verify that the Client is able to use Renew operation as keep alive for Pull Point subscription.

Pre-Requisite:

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

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

1. Client invokes CreatePullPointSubscription message.

2. Device responds with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.

3. Client invokes Renew message to valid address recived in CreatePullPointSubscriptionResponse message for the created Pull Point subscription with valid address recived in CreatePullPointSubscriptionResponse message.


Test Result:

PASS -

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

- Client Renew request in Test Procedure fulfills the following requirements:

  - [S1] soapenv:Body element has child element wsnt:Renew AND

- Device response on the Renew request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND
• [S3] soapenv:Body element has child element wsnt:RenewResponse AND

• There is a Device response on the CreatePullPointSubscription request in Test Procedure fulfills the following requirements:

• [S4] It has HTTP 200 response code AND

• [S5] It received for the same Device as for the Client Renew request AND

• [S6] It received before the Client Renew request AND

• [S7] It contains tev:SubscriptionReference|wsa:Address element which is equal to HTTP address that was used to send the Renew request.

FAIL -

• The Client failed PASS criteria.

5.5.4 PULL MESSAGES AS KEEP ALIVE

Test Label: Advanced Pull Point Event Handling - Pull Messages as Keep Alive

Test Case ID: KEEPALIVEFORPULLPOINTEVENTHANDLING-2

Feature Under Test: Pull Messages as Keep Alive (KeepAliveForPullPointEventHandling_PullMessagesAsKeepAlive)

Profile A Normative Reference: Mandatory

Profile C Normative Reference: Mandatory

Profile S Normative Reference: Conditional

Profile Q Normative Reference: Optional

Profile G Normative Reference: Conditional

Profile T Normative Reference: Optional

Test Purpose: To verify that the Client is able to use PullMessages operation as keep alive for Pull Point subscription.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with CreatePullPointSubscription operations with tev:InitialTerminationTime element present.
Test Procedure (expected to be reflected in network trace file):

1. Client invokes `CreatePullPointSubscription` message.

2. Device responds with code HTTP 200 OK and `CreatePullPointSubscriptionResponse` message without `tev:InitialTerminationTime` element.

Test Result:

PASS -

- Client `CreatePullPointSubscription` request messages are valid according to XML Schemas listed in Namespaces AND
- Client `CreatePullPointSubscription` request in Test Procedure fulfills the following requirements:
  - [S1] `soapenv:Body` element has child element `tev:CreatePullPointSubscription` AND
  - [S2] It does not contain `tev:InitialTerminationTime` element AND
- Device response on the `CreatePullPointSubscription` request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND

FAIL -

- The Client failed PASS criteria.

5.6 Discovery Test Cases

5.6.1 Feature Level Requirement:

Validated Feature: Discovery (Discovery)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile S Requirement: Conditional

Profile C Requirement: Conditional
**Profile G Requirement:** Conditional

**Profile A Requirement:** Mandatory

**Profile Q Requirement:** Mandatory

**Profile T Requirement:** Mandatory

**Profile D Requirement:** Mandatory

**Profile M Requirement:** Mandatory

### 5.6.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IP address 239.255.255.250 and port 3702 to locate services on a local network.

2. Client is considered as supporting Discovery if the following conditions are met:
   - Probe request detected AND at least one ProbeMatch response detected

3. Client is considered as NOT supporting Discovery if the following is TRUE:
   - No Valid Device Response to Probe request.

### 5.6.3 WS-DISCOVERY

**Test Label:** Discovery - WS-Discovery

**Test Case ID:** DISCOVERY-1

**Feature Under Test:** WS-Discovery (Discovery_WSDiscovery)

**Profile S Normative Reference:** Conditional

**Profile G Normative Reference:** Conditional

**Profile C Normative Reference:** Conditional

**Profile Q Normative Reference:** Mandatory

**Profile A Normative Reference:** Mandatory

**Profile T Normative Reference:** Mandatory

**Profile D Normative Reference:** Mandatory

**Profile M Normative Reference:** Mandatory
Test Purpose: To verify that Client is able to send Probe request and receive ProbeMatch response from Device.

Pre-Requisite:

- The Network Trace Capture files contain at least one Client Probe request to multicast IP address and one ProbeMatch response from Device directly to the Client.

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

1. Client invokes Probe request message to multicast IP address 239.255.255.250 and port 3702.
2. Device sends ProbeMatch message directly to the Client.

Test Result:

PASS -

- Client Probe request messages are valid according to XML Schemas listed in Namespaces AND
- Client Probe request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<Action>" tag after the "<Header>" tag AND
  - [S2] "<Action>" includes URL address which ends with "Probe" value AND
  - [S3] Client request contains "<MessageID>" with non-empty string value AND
  - [S4] Client request contains "<Probe>" tag after the "<Body>" tag AND

FAIL -

- The Client failed PASS criteria.

5.7  Device Discovery Type Filter Test Cases

5.7.1  Feature Level Requirement:

Validated Feature: Device Discovery Type Filter (DeviceDiscoveryTypeFilter)

Check Condition based on Device Features: Device Discovery Type is supported by Device.

Required Number of Devices: 3
Profile S Requirement: None

Profile A Requirement: Mandatory

Profile C Requirement: Conditional

Profile D Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile G Requirement: Conditional

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.7.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter is equal to tds:Device or with skipped Types filter.

2. Client is considered as supporting Device Discovery Type if the following conditions are met:
   - **Probe** Client message that fulfills the following requirement is detected:
     - Types filter is equal to tds:Device or empty or skipped AND
     - Probe is sent to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] AND
     - Probe is sent to UDP port 3702 AND
     - There is **ProbeMatch** Device message that correspond to Client **Probe**.

3. Client is considered as NOT supporting Device Discovery Type if the following is TRUE:
   - No valid Device **ProbeMatch** message that is correspond to Client **Probe** message.

5.7.3 DEVICE DISCOVERY TYPE FILTER

**Test Label:** Discovery - Device Discovery Type Filter

**Test Case ID:** DEVCEDISCOVERYTYPEFILTER-1

**Feature Under Test:** Device Discovery Type Filter

(DeviceDiscoveryTypeFilter_DeviceDiscoveryFilter)
**Profile S Normative Reference:** None

**Profile G Normative Reference:** Mandatory

**Profile C Normative Reference:** Mandatory

**Profile Q Normative Reference:** Mandatory

**Profile A Normative Reference:** Mandatory

**Profile T Normative Reference:** Mandatory

**Profile D Normative Reference:** Mandatory

**Profile M Normative Reference:** Mandatory

**Test Purpose:** To verify that Client is able to discover devices with Device Discovery Type.

**Pre-Requisite:**

- The Network Trace Capture files contains at least one Client Probe message that does not filter out devices with Device Discovery Type that is sent to multicast WS-Discovery address.

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

1. Client invokes Probe request message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with **Types** = tds:Device.

2. Device sends ProbeMatch message to the Client.

**Test Result:**

**PASS -**

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

- Client **Probe** request in Test Procedure fulfills the following requirements:
  
  - [S1] It is sent to 239.255.255.250 IPv4 address OR [FF02::C] IPv6 address AND
  
  - [S2] It is sent to 3702 UDP port AND
  
  - [S3] soapenv:Envelope/soapenv:Header element has child element **wsadis:Action** AND
  
  - [S4] **wsadis:Action** includes URL address which ends with "Probe" value AND
• [S5] soapenv:Envelope/soapenv:Header element has child element wsadis:MessageID with non-empty string value AND

• [S6] soapenv:Body element has child element d:Probe AND

• [S7] IF d:Probe element has child element d:Types THEN it has value is equal to tds:Device OR empty string value AND

• [S8] There is Device ProbeMatches message in test procedure that fulfills the following requirements:
  • [S9] soapenv:Body element has child element d:ProbeMatches AND
  • [S10] soapenv:Envelope/soapenv:Header/wsadis:RelatesTo element value is equal to soapenv:Envelope/soapenv:Header/wsadis:MessageID value in Probe message AND

PASS WITH WARNING -
  • d:Probe/d:Types element is skipped OR
  • d:Probe/d:Types element has empty string value.

FAIL -
  • The Client failed PASS criteria.

5.8 User Handling Test Cases

5.8.1 Feature Level Requirement:

Validated Feature: User Handling (UserHandling)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile Q Requirement: Mandatory

Profile S Requirement: Conditional

Profile C Requirement: Conditional

Profile G Requirement: Conditional
Profile T Requirement: Conditional

Profile D Requirement: Conditional

5.8.2 Expected Scenarios Under Test:

1. Client connects to Device to create, list, modify and delete users.

2. Client is considered as supporting User Handling if the following conditions are met:
   - Client is able to create users on Device using the CreateUsers operation AND
   - Client is able to list existing users of Device using the GetUsers operation AND
   - Client is able to modify users on Device using the SetUser operation AND
   - Client is able to delete users from Device using the DeleteUsers operation.

3. Client is considered as NOT supporting System if ANY of the following is TRUE:
   - No Valid Device Response to CreateUsers request (except SOAP fault: soapenv:Receiver/ter:Action/ter:TooManyUsers) OR
   - No Valid Device Response to GetUsers request OR
   - No Valid Device Response to SetUser request (except SOAP fault: soapenv:Sender/ter:InvalidArgVal/ter:FixedUser) OR

5.8.3 CREATE USERS

Test Label: User Handling - CreateUsers

Test Case ID: USERHANDLING-1

Feature Under Test: Create Users (UserHandling_CreateUsers)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory
Profile T Normative Reference: Conditional

Profile D Normative Reference: Conditional

Test Purpose: To verify that Client is able to create users on Device using the CreateUsers operation.

Pre-Requisite:

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

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

1. Client invokes CreateUsers request message to create new users and corresponding credentials on Device.

2. Device responds with code HTTP 200 OK and CreateUsersResponse message.

Test Result:

PASS -

• Client CreateUsers request messages are valid according to XML Schemas listed in Namespaces AND

• Client CreateUsers request in Test Procedure fulfills the following requirements:

  • [S1] Client request contains "<CreateUsers>" tag after the "<Body>" tag AND

  • [S2] "<CreateUsers>" includes tag: "<User>" AND

  • [S3] "<User>" includes tag: "<Username>" with non-empty string value AND

  • [S4] "<User>" includes tag: "<Password>" with non-empty string value AND


FAIL -

• The Client failed PASS criteria.

5.8.4 GET USERS

Test Label: User Handling - GetUsers
Test Case ID: USERHANDLING-2

Feature Under Test: Get Users (UserHandling_GetUsers)

Profile S Normative Reference: Conditional
Profile G Normative Reference: Conditional
Profile C Normative Reference: Conditional
Profile Q Normative Reference: Mandatory
Profile A Normative Reference: Mandatory
Profile T Normative Reference: Conditional
Profile D Normative Reference: Conditional

Test Purpose: To verify that Client is able to list existing users of Device using the GetUsers operation.

Pre-Requisite:

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

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

1. Client invokes GetUsers request message to list registered users and their user levels.
2. Device responds with code HTTP 200 OK and GetUsersResponse message.

Test Result:

PASS -

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

- Client GetUsers request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetUsers>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND

FAIL -
5.8.5 SET USER

Test Label: User Handling - SetUser

Test Case ID: USERHANDLING-3

Feature Under Test: Set User (UserHandling_SetUser)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Mandatory

Profile A Normative Reference: Mandatory

Profile T Normative Reference: Conditional

Profile D Normative Reference: Conditional

Test Purpose: To verify that Client is able to modify users on Device using the SetUser operation.

Pre-Requisite:

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

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

1. Client invokes SetUser request message to update the authentication settings on Device.
2. Device responds with code HTTP 200 OK and SetUserResponse message.

Test Result:

PASS -

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

- Client SetUser request in Test Procedure fulfills the following requirements:
  
  - [S1] Client request contains "<SetUser>" tag after the "<Body>" tag AND
  
  - [S2] "<SetUser>" includes tag: "<User>" AND
• [S3] "<User>" includes tag: "<Username>" with non-empty string value AND


FAIL -

• The Client failed PASS criteria.

5.8.6 DELETE USERS

Test Label: User Handling - DeleteUsers

Test Case ID: USERHANDLING-4

Feature Under Test: Delete Users (UserHandling_DeleteUsers)

Profile S Normative Reference: Conditional
Profile G Normative Reference: Conditional
Profile C Normative Reference: Conditional
Profile Q Normative Reference: Mandatory
Profile A Normative Reference: Mandatory
Profile T Normative Reference: Conditional
Profile D Normative Reference: Conditional

Test Purpose: To verify that Client is able to delete users from Device using the DeleteUsers operation.

Pre-Requisite:

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

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

1. Client invokes DeleteUsers request message to delete specific users from Device.

2. Device responds with code HTTP 200 OK and DeleteUsersResponse message.

Test Result:
PASS -

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

- Client **DeleteUsers** request in Test Procedure fulfills the following requirements:
  
  - [S1] Client request contains "<DeleteUsers>" tag after the "<Body>" tag AND
  
  - [S2] "<DeleteUsers>" includes tag: "<Username>" with non-empty string value AND
  

FAIL -

- The Client failed PASS criteria.

5.9 Get Credential Capabilities Test Cases

5.9.1 Feature Level Normative Reference:

**Validated Feature**: Get Credential Capabilities (GetCredentialCapabilities)

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

**Required Number of Devices**: 3

**Profile A Requirement**: Mandatory

5.9.2 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:
5.9.3 GET SERVICE CAPABILITIES

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

**Test Case ID:** GETCREDENTIALCAPABILITIES-1

**Feature Under Test:** Get Credential Service Capabilities

(GetCredentialCapabilities_GetCredentialServiceCapabilities)

**Profile A Normative Reference:** Optional

**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.

5.10 Get Credential List Test Cases

5.10.1 Feature Level Normative Reference:

Validated Feature: Get Credential List (GetCredentialList)

Check Condition based on Device Features: Credential Service is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

5.10.2 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 received in GetCredentialInfoList response in tcr:NextStartReference element OR
   • GetCredentialList request contains tcr:StartReference element value that was not received in GetCredentialList response in tcr:NextStartReference element OR
   • Complete Credentials list was not received.

5.10.3 LISTING OF CREDENTIALS

Test Label: Get Credential List - Listing of Credentials
Test Case ID: GETCREDENTIALLIST-1

Feature Under Test: Get Credential List (GetCredentialList_GetCredentialListRequest)

Profile A Normative Reference: Optional

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.


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.

5.10.4 LISTING OF CREDENTIAL INFO

Test Label: Get Credential List - Listing of Credential Info

Test Case ID: GETCREDENTIALLIST-2

Feature Under Test: Get Credential Info List (GetCredentialList_GetCredentialInfoListRequest)

Profile A Normative Reference: Mandatory

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.
5.11 Get Credential Details Test Cases

5.11.1 Feature Level Normative Reference:

Validated Feature: Get Credential Details (GetCredentialDetails)

Check Condition based on Device Features: Credential Service is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

5.11.2 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.

5.11.3 GET CREDENTIALS

Test Label: Get Credential Details - Get Credentials

Test Case ID: GETCREDENTIALDETAILS-1

Feature Under Test: Get Credentials (GetCredentialDetails:GetCredentialDetailsRequest)

Profile A Normative Reference: Mandatory

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.

5.12 Configure Credentials Test Cases

5.12.1 Feature Level Normative Reference:

**Validated Feature:** Configure Credentials (ConfigureCredentials)

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

**Required Number of Devices:** 3

**Profile A Requirement:** Mandatory

5.12.2 Expected Scenarios Under Test:

1. Client supports get_credential_capabilities feature.
2. Client gets supported format types of a specified identifier type using the `GetSupportedFormatTypes` operation to use it for `CreateCredential` operation and `ModifyCredential` operation.

3. Client creates credentials on a Device using the `CreateCredential` operation.

4. Client modifies credentials on a Device using the `ModifyCredential` operation.

5. Client deletes credentials from a Device using the `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 the `GetServiceCapabilities` operation or the `GetServices` operation AND
   - Client is able to get supported format types of a specified identifier type using the `GetSupportedFormatTypes` operation AND
   - Client is able to create credential using the `CreateCredential` operation AND
   - Client is able to modify credential using the `ModifyCredential` operation AND
   - Client is able to delete credential using the `DeleteCredential` operation.

7. Client is considered as NOT supporting Configure Credentials if ANY of the following is TRUE:
   - No valid responses for the `GetSupportedFormatTypes` request OR
   - No valid responses for the `CreateCredential` request OR
   - No valid responses for the `ModifyCredential` request OR
   - No valid responses for the `DeleteCredential` request.

5.12.3 GET SUPPORTED FORMAT TYPES

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

**Test Case ID:** CONFIGURECREDENTIALS-1

**Feature Under Test:** Get Supported Format Types
(ConfigureCredentials_GetSupportedFormatTypes)

**Profile A Normative Reference:** Mandatory

**Profile D Normative Reference:** Conditional
**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.

**5.12.4 CREATE CREDENTIAL**

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

**Test Case ID:** CONFIGURECREDENTIALS-2
Feature Under Test: Create Credential (ConfigureCredentials_CreateCredential)

Profile A Normative Reference: Mandatory

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.


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.

5.12.5 MODIFY CREDENTIAL

Test Label: Configure Credentials - Modify Credential

Test Case ID: CONFIGURECREDENTIALS-3

Feature Under Test: Modify Credential (ConfigureCredentials_ModifyCredential)
Profile A Normative Reference: Mandatory

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

Pre-Requisite:

- The Network Trace Capture files contain 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:ValidTo tcr: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
FAIL -
  • The Client failed PASS criteria.

5.12.6 DELETE CREDENTIAL

Test Label: Configure Credentials - Delete Credential

Test Case ID: CONFIGURECREDENTIALS-4

Feature Under Test: Delete Credential (ConfigureCredentials_DeleteCredential)

Profile A Normative Reference: Mandatory

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.

5.13 Credential Configuration and State Notifications Test Cases

5.13.1 Feature Level Normative Reference:

**Validated Feature:** Credentials Notifications (CredentialsNotifications)

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

**Required Number of Devices:** 3

**Profile A Requirement:** Mandatory

5.13.2 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.

5.14 Get Schedule List Test Cases

5.14.1 Feature Level Normative Reference:

Validated Feature: Get Schedule List (GetScheduleList)

Check Condition based on Device Features: Schedule Service is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

5.14.2 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 received in GetScheduleInfoList response in tsc:NextStartReference element OR

• GetScheduleList request contains tsc:StartReference element value that was not received in GetScheduleList response in tsc:NextStartReference element OR

• Complete Schedules list was not received.

5.14.3 LISTING OF SCHEDULES

Test Label: Get Schedule List - Listing of Schedules

Test Case ID: GETSCHEDULELIST-1

Feature Under Test: Get Schedule List (GetScheduleList_GetScheduleListRequest)

Profile A Normative Reference: Optional

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.


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.


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.

5.14.4 LISTING OF SCHEDULE INFO

Test Label: Get Schedule List - Listing of Schedule Info

Test Case ID: GETSCHEDULELIST-2

Feature Under Test: Get Schedules Info List (GetScheduleList_GetSchedulesInfoListRequest)

Profile A Normative Reference: Mandatory

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.


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.


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.

5.15 Schedule Configuration Notifications Test Cases

5.15.1 Feature Level Normative Reference:

**Validated Feature:** Schedules Notifications (SchedulesNotifications)

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

**Required Number of Devices:** 3

**Profile A Requirement:** Mandatory

5.15.2 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.

5.16 Get Access Profile List Test Cases

5.16.1 Feature Level Normative Reference:

Validated Feature: Get Access Profile List GetAccessProfileList

Check Condition based on Device Features: Access Rules Service is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

5.16.2 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 received in GetAccessProfileInfoList response in tsc:NextStartReference element OR
   • GetAccessProfileList request contains tsc:StartReference element value that was not received in GetAccessProfileList response in tsc:NextStartReference element OR
5.16.3 LISTING OF ACCESS PROFILES

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

Test Case ID: GETACCESSPROFILELIST-1

Feature Under Test: Get Access Profile List (GetAccessProfileList_GetAccessProfileListRequest)

Profile A Normative Reference: Optional

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.


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.


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.

5.16.4 LISTING OF ACCESSPROFILE INFO

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

Test Case ID: GETACCESSPROFILELIST-2

Feature Under Test: Get Access Profile Info List
(GetAccessProfileList_GetAccessProfileInfoListRequest)

Profile A Normative Reference: Mandatory

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 \texttt{GetAccessProfileInfoList} request message with skipped \texttt{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 \texttt{GetAccessProfileInfoListResponse} message.

3. If \texttt{GetAccessProfileInfoListResponse} message contains \texttt{tar:NextStartReference} element Client invokes \texttt{GetAccessProfileInfoList} request message with \texttt{tar:StartReference} element equal to value of \texttt{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 \texttt{GetAccessProfileInfoListResponse} message contains \texttt{tar:NextStartReference} element.

Test Result:

PASS -

• Client \texttt{GetAccessProfileInfoList} request messages are valid according to XML Schemas listed in Namespaces AND

• First Client \texttt{GetAccessProfileInfoList} request in Test Procedure fulfills the following requirements:
  
  • [S1] \texttt{soapenv:Body} element has child element \texttt{tar:GetAccessProfileInfoList} AND

  • [S2] It does not contain \texttt{tar:StartReference} element AND

• If response on previous \texttt{GetAccessProfileInfoList} request contains \texttt{tcr:NextStartReference} element each next Client \texttt{GetAccessProfileInfoList} requests in Test Procedure fulfills the following requirements (else skip the checks):

  • [S3] \texttt{soapenv:Body} element has child element \texttt{tar:GetAccessProfileInfoList} AND

  • [S4] It contains \texttt{tar:StartReference} element equal to \texttt{tar:NextStartReference} element from response on previous \texttt{GetAccessProfileInfoList} request AND

• Device responses on the each \texttt{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.

5.17 Access Profile Configuration Notifications Test Cases

5.17.1 Feature Level Normative Reference:

Validated Feature: Access Profile Notifications (AccessProfileNotifications)

Check Condition based on Device Features: Access Rules Service is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

5.17.2 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_WSSecurityNotification` 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.
6 Test Cases for Profile Conditional Features

6.1 Network Configuration Test Cases

6.1.1 Feature Level Requirement:

**Validated Feature:** Network Configuration (NetworkConfiguration)

**Check Condition based on Device Features:** None

**Required Number of Devices:** 3

**Profile A Requirement:** Conditional

**Profile C Requirement:** Conditional

**Profile D Requirement:** Mandatory

**Profile G Requirement:** Conditional

**Profile Q Requirement:** Conditional

**Profile S Requirement:** Conditional

**Profile T Requirement:** Mandatory

**Profile M Requirement:** Mandatory

6.1.2 Expected Scenarios Under Test:

1. Client connects to Device to configure network settings.

2. Client is considered as supporting Network Configuration if the following conditions are met:
   
   - Client is able to list network interfaces of Device using the GetNetworkInterfaces operation AND
   
   - Client is able to set network interfaces of Device using the SetNetworkInterfaces operation AND
   
   - Client is able to list default gateway of Device using the GetNetworkDefaultGateway operation AND
   
   - Client is able set default gateway of Device using the SetNetworkDefaultGateway operation.
3. Client is considered as NOT supporting Network Configuration if ANY of the following is TRUE:

- No Valid Device Response to GetNetworkInterfaces request OR
- No Valid Device Response to SetNetworkInterfaces request OR
- No Valid Device Response to GetNetworkDefaultGateway request OR
- No Valid Device Response to SetNetworkDefaultGateway request.

6.1.3 GET NETWORK INTERFACES

Test Label: Network Configuration - Get Network Interfaces

Test Case ID: NETWORKCONFIGURATION-1

Feature Under Test: Get Network Interfaces (NetworkConfiguration_GetNetworkInterfaces)

Profile S Normative Reference: Conditional
Profile G Normative Reference: Conditional
Profile C Normative Reference: Conditional
Profile Q Normative Reference: Conditional
Profile A Normative Reference: Conditional
Profile T Normative Reference: Mandatory
Profile D Normative Reference: Mandatory
Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to list network interfaces of Device using the GetNetworkInterfaces operation.

Pre-Requisite:

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

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

1. Client invokes GetNetworkInterfaces request message to get network interface configuration from Device.
2. Device responds with code HTTP 200 OK and GetNetworkInterfacesResponse message.

Test Result:

PASS -

• Client GetNetworkInterfaces request messages are valid according to XML Schemas listed in Namespaces AND

• Client GetNetworkInterfaces request in Test Procedure fulfills the following requirements:

  • [S1] Client request contains "<GetNetworkInterfaces>" tag after the "<Body>" tag AND

  • [S2] Device response contains "HTTP/* 200 OK" AND


FAIL -

• The Client failed PASS criteria.

6.1.4 SET NETWORK INTERFACES

Test Label: Network Configuration - Set Network Interfaces

Test Case ID: NETWORKCONFIGURATION-2

Feature Under Test: Set Network Interfaces (NetworkConfiguration_SetNetworkInterfaces)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional

Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to set network interfaces of Device using the SetNetworkInterfaces operation.
Pre-Requisite:

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

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

1. Client invokes SetNetworkInterfaces request message to set the network interface configuration on Device.

2. Device responds with code HTTP 200 OK and SetNetworkInterfacesResponse message.

Test Result:

PASS -

• Client SetNetworkInterfaces request messages are valid according to XML Schemas listed in Namespaces AND

• Client SetNetworkInterfaces request in Test Procedure fulfills the following requirements:

  • [S1] Client request contains "<SetNetworkInterfaces>" tag after the "<Body>" tag AND

  • [S2] "<SetNetworkInterfaces>" includes tag: "<InterfaceToken>" with non-empty string value of specific token AND

  • [S4] Device response contains "HTTP/* 200 OK" AND


FAIL -

• The Client failed PASS criteria.

6.1.5 GET NETWORK DEFAULT GATEWAY

Test Label: Network Configuration - Get Network Default Gateway

Test Case ID: NETWORKCONFIGURATION-3

Feature Under Test: Get Network Default Gateway
(NetworkConfiguration_GetNetworkDefaultGateway)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Conditional
Profile C Normative Reference: Conditional

Profile Q Normative Reference: Conditional

Profile A Normative Reference: Conditional

Profile T Normative Reference: Mandatory

Profile D Normative Reference: Mandatory

Profile M Normative Reference: Mandatory

Test Purpose: To verify that Client is able to list default gateway of Device using the GetNetworkDefaultGateway operation.

Pre-Requisite:

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

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

1. Client invokes GetNetworkDefaultGateway request message to get the default gateway settings from Device.


Test Result:

PASS -

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

- Client GetNetworkDefaultGateway request in Test Procedure fulfills the following requirements:

  - [S1] Client request contains "<GetNetworkDefaultGateway>" tag after the "<Body>" tag AND

  - [S2] Device response contains "HTTP/* 200 OK" AND


FAIL -

- The Client failed PASS criteria.
6.1.6 SET NETWORK DEFAULT GATEWAY

Test Label: Network Configuration - Set Network Default Gateway

Test Case ID: NETWORKCONFIGURATION-4

Test Purpose: To verify that Client is able to set default gateway of Device using the SetNetworkDefaultGateway operation.

Pre-Requisite:

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

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

1. Client invokes SetNetworkDefaultGateway request message to set the default gateway settings on Device.

2. Device responds with code HTTP 200 OK and SetNetworkDefaultGatewayResponse message.

Test Result:

PASS -

- Client SetNetworkDefaultGateway request messages are valid according to XML Schemas listed in Namespaces AND
Client \texttt{SetNetworkDefaultGateway} request in Test Procedure fulfills the following requirements:

- [S1] Client request contains "\texttt{<SetNetworkDefaultGateway>}" tag after the "\texttt{<Body>}" tag AND

- [S2] "\texttt{<SetNetworkDefaultGateway>}" includes tag: EITHER "\texttt{<IPv4Address>}" OR "\texttt{<IPv6Address>}" with specific IP address value AND

- [S3] Device response contains "HTTP/* 200 OK" AND


\textbf{FAIL -}

- The Client failed PASS criteria.

\section{6.2 System Test Cases}

\subsection{6.2.1 Feature Level Requirement:}

\textbf{Validated Feature:} System (System)

\textbf{Check Condition based on Device Features:} None

\textbf{Required Number of Devices:} 3

\textbf{Profile A Requirement:} Conditional

\textbf{Profile C Requirement:} Conditional

\textbf{Profile G Requirement:} Conditional

\textbf{Profile Q Requirement:} Conditional

\textbf{Profile S Requirement:} Conditional

\textbf{Profile T Requirement:} Conditional

\textbf{Profile D Requirement:} Conditional

\subsection{6.2.2 Expected Scenarios Under Test:}

1. Client connects to Device to get information, such as manufacturer, model, firmware version and etc.

2. Client is considered as supporting System if the following conditions are met:
• Client is able to list Device information using the GetDeviceInformation operation.

3. Client is considered as NOT supporting System if ANY of the following is TRUE:
   • No Valid Device Response to GetDeviceInformation request.

6.2.3 GET DEVICE INFORMATION

Test Label: System - Get Device Information

Test Case ID: SYSTEM-1

Feature Under Test: Get Device Information (System_GetDeviceInformation)

Profile S Normative Reference: Conditional
Profile G Normative Reference: Conditional
Profile C Normative Reference: Conditional
Profile Q Normative Reference: Conditional
Profile A Normative Reference: Conditional
Profile T Normative Reference: Conditional
Profile D Normative Reference: Conditional

Test Purpose: To verify that Client is able to list Device information using the GetDeviceInformation operation.

Pre-Requisite:
   • The Network Trace Capture files contains at least one Conversation between Client and Device with GetDeviceInformation operation present.

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

1. Client invokes GetDeviceInformation request message to list Device information.

Test Result:

PASS -
   • Client GetDeviceInformation request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetDeviceInformation** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetDeviceInformation>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND

FAIL -
- The Client failed PASS criteria.

### 6.3  IP Address Filtering Test Cases

#### 6.3.1 Feature Level Requirement:

**Validated Feature**: IP Address Filtering (IPAddressFiltering)

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

**Required Number of Devices**: 1

**Profile S Requirement**: Conditional

**Profile C Requirement**: Conditional

**Profile A Requirement**: Conditional

#### 6.3.2 Expected Scenarios Under Test:

1. Client connects to Device to manage IP address filters.

2. Client is considered as supporting IP Address Filtering if the following conditions are met:
   - Client is able to get the IP address filter settings from Device using the GetIPAddressFilter operation AND
   - Client is able to set the IP address filter settings on Device using the SetIPAddressFilter operation AND
   - Client is able to add the IP address filter settings to Device using the AddIPAddressFilter operation AND
   - Client is able to delete the IP address filter settings from Device using the RemoveIPAddressFilter operation.
• **NOTE:** Requests `SetIPAddressFilter`, `AddIPAddressFilter` and `RemoveIPAddressFilter` are permitted to use the IPv4 OR IPv6 protocol settings.

3. Client is considered as NOT supporting IP Address Filtering if ANY of the following is TRUE:
   • No Valid Device Response to `GetIPAddressFilter` request OR
   • No Valid Device Response to `SetIPAddressFilter` request OR
   • No Valid Device Response to `AddIPAddressFilter` request OR
   • No Valid Device Response to `RemoveIPAddressFilter` request.

   • **NOTE:** Requests `SetIPAddressFilter`, `AddIPAddressFilter` and `RemoveIPAddressFilter` should be deemed as failed if both IPv4 AND IPv6 protocol settings have No Valid Device Responses.

### 6.3.3 GET IP ADDRESS FILTER

**Test Label:** IP Address Filtering - `GetIPAddressFilter`

**Test Case ID:** IPADDRESSFILTERING-1

**Feature Under Test:** Get Ip Address Filter (`IPAddressFiltering_GetIpAddressFilter`)

**Profile S Normative Reference:** Conditional

**Profile G Normative Reference:** Optional

**Profile C Normative Reference:** Conditional

**Profile A Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to get the IP address filter settings from Device using the `GetIPAddressFilter` operation.

**Pre-Requisite:**

• The Network Trace Capture files contains at least one Conversation between Client and Device with `GetIPAddressFilter` operation present.

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

1. Client invokes `GetIPAddressFilter` request message to get the IP address filter settings from Device.

2. Device responds with code HTTP 200 OK and `GetIPAddressFilterResponse` message.
Test Result:

PASS -

- Client `GetIPAddressFilter` request messages are valid according to XML Schemas listed in Namespaces AND

- Client `GetIPAddressFilter` request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetIPAddressFilter>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND

FAIL -

- The Client failed PASS criteria.

6.3.4 SET IPv4 ADDRESS FILTER

Test Label: IP Address Filtering - SetIPv4AddressFilter

Test Case ID: IPADDRESSFILTERING-2

Feature Under Test: Set IPv4 Address Filter (IPAddressFiltering_SetIpV4AddressFilter)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Optional

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to set the IP address filter settings on Device using the SetIPAddressFilter operation.

Pre-Requisite:

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

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

1. Client invokes SetIPAddressFilter request message to set the IP address filter settings on Device.
2. Device responds with code HTTP 200 OK and SetIPAddressFilterResponse message.

Test Result:

NOTE: If Client SetIPAddressFilter request message does not contain "<IPv4Address>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

• Client SetIPAddressFilter request messages are valid according to XML Schemas listed in Namespaces AND

• Client SetIPAddressFilter request in Test Procedure fulfills the following requirements:
  • [S1] Client request contains "<SetIPAddressFilter>" tag after the "<Body>" tag AND
  • [S3] "<SetIPAddressFilter>" includes tag: "<IPv4Address>" AND
  • [S4] "<IPv4Address>" includes tag: "<Address>" with specific IPv4 address value AND
  • [S5] "<IPv4Address>" includes tag: "<PrefixLength>" with value range from "0" to "32" AND
  • [S6] Device response contains "HTTP/* 200 OK" AND

FAIL -

• The Client failed PASS criteria.

6.3.5 SET IPv6 ADDRESS FILTER

Test Label: IP Address Filtering - SetIPv6AddressFilter

Test Case ID: IPADDRESSFILTERING-3

Feature Under Test: Set IPv6 Address Filter (IPAddressFiltering_SetIpV6AddressFilter)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Optional

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to set the IP address filter settings on Device using the SetIPAddressFilter operation.
Pre-Requisite:

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

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

1. Client invokes SetIPAddressFilter request message to set the IP address filter settings on Device.

2. Device responds with code HTTP 200 OK and SetIPAddressFilterResponse message.

Test Result:

NOTE: If Client SetIPAddressFilter request message does not contain "<IPv6Address>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

• Client SetIPAddressFilter request messages are valid according to XML Schemas listed in Namespaces AND

• Client SetIPAddressFilter request in Test Procedure fulfills the following requirements:

  • [S1] Client request contains "<SetIPAddressFilter>" tag after the "<Body>" tag AND
  • [S3] "<SetIPAddressFilter>" includes tag: "<IPv6Address>" AND
  • [S4] "<IPv6Address>" includes tag: "<Address>" with specific IPv6 address value AND
  • [S5] "<IPv6Address>" includes tag: "<PrefixLength>" with value range from "0" to "128" AND
  • [S6] Device response contains "HTTP/* 200 OK" AND

FAIL -

• The Client failed PASS criteria.

6.3.6 ADD IPv4 ADDRESS FILTER

Test Label: IP Address Filtering - AddIPv4AddressFilter

Test Case ID: IPAddressFiltering-4

Feature Under Test: Add IPv4 Address Filter (IPAddressFiltering_AddIPv4AddressFilter)
Profile S Normative Reference: Conditional

Profile G Normative Reference: Optional

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to add the IP address filter to Device using the AddIPAddressFilter operation.

Pre-Requisite:

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

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

1. Client invokes AddIPAddressFilter request message to add the IP address filter on Device.

2. Device responds with code HTTP 200 OK and AddIPAddressFilterResponse message.

Test Result:

NOTE: If Client AddIPAddressFilter request message does not contain "<IPv4Address>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

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

- Client AddIPAddressFilter request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<AddIPAddressFilter>" tag after the "<Body>" tag AND
  - [S3] "<AddIPAddressFilter>" includes tag: "<IPv4Address>" AND
  - [S4] "<IPv4Address>" includes tag: "<Address>" with specific IPv4 address value AND
  - [S5] "<IPv4Address>" includes tag: "<PrefixLength>" with value range from "0" to "32" AND
  - [S6] Device response contains "HTTP/* 200 OK" AND

FAIL -

- The Client failed PASS criteria.
6.3.7 ADD IPv6 ADDRESS FILTER

Test Label: IP Address Filtering - AddIPv6AddressFilter

Test Case ID: IPADDRESSFILTERING-5

Feature Under Test: Add IPv6 Address Filter (IPAddressFiltering_AddIpV6AddressFilter)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Optional

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to add the IP address filter to Device using the AddIPAddressFilter operation.

Pre-Requisite:

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

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

1. Client invokes AddIPAddressFilter request message to add the IP address filter on Device.

2. Device responds with code HTTP 200 OK and AddIPAddressFilterResponse message.

Test Result:

NOTE: If Client AddIPAddressFilter request message does not contain "<IPv6Address>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

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

- Client AddIPAddressFilter request in Test Procedure fulfills the following requirements:

  - [S1] Client request contains "<AddIPAddressFilter>" tag after the "<Body>" tag AND

  - [S3] "<AddIPAddressFilter>" includes tag: "<IPv6Address>" AND

  - [S4] "<IPv6Address>" includes tag: "<Address>" with specific IPv6 address value AND
• [S5] "<IPv6Address>" includes tag: "<PrefixLength>" with value range from "0" to "128" AND

• [S6] Device response contains "HTTP/* 200 OK" AND


FAIL -

• The Client failed PASS criteria.

6.3.8 REMOVE IPv4 ADDRESS FILTER

Test Label: IP Address Filtering - RemoveIPv4AddressFilter

Test Case ID: IPADDRESSFILTERING-6

Feature Under Test: Remove IPv4 Address Filter (IPAddressFiltering_RemoveIpV4AddressFilter)

Profile S Normative Reference: Conditional

Profile G Normative Reference: Optional

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to delete the IP address filter from Device using the RemoveIPAddressFilter operation.

Pre-Requisite:

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

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

1. Client invokes RemoveIPAddressFilter request message to delete the IP address filter from Device.

2. Device responds with code HTTP 200 OK and RemoveIPAddressFilterResponse message.

Test Result:

NOTE: If Client RemoveIPAddressFilter request message does not contain "<IPv4Address>" tag then Test shall be deemed as "NOT DETECTED".
PASS -

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

- Client **RemoveIPAddressFilter** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<RemoveIPAddressFilter>" tag after the "<Body>" tag AND
  - [S3] "<RemoveIPAddressFilter>" includes tag: "<IPv4Address>" AND
  - [S4] "<IPv4Address>" includes tag: "<Address>" with specific IPv4 address value AND
  - [S5] "<IPv4Address>" includes tag: "<PrefixLength>" with value range from "0" to "32" AND
  - [S6] Device response contains "HTTP/* 200 OK" AND

FAIL -

- The Client failed PASS criteria.

6.3.9 REMOVE IPv6 ADDRESS FILTER

**Test Label:** IP Address Filtering - RemoveIPv6AddressFilter

**Test Case ID:** IPAddressFiltering-7

**Feature Under Test:** Remove IPv6 Address Filter (IPAddressFiltering_RemoveIPv6AddressFilter)

**Profile S Normative Reference:** Conditional

**Profile G Normative Reference:** Optional

**Profile C Normative Reference:** Conditional

**Profile A Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to delete the IP address filter from Device using the RemoveIPAddressFilter operation.

**Pre-Requisite:**

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

**Test Procedure (expected to be reflected in network trace file):**
1. Client invokes RemoveIPAddressFilter request message to delete the IP address filter from Device.

2. Device responds with code HTTP 200 OK and RemoveIPAddressFilterResponse message.

Test Result:

NOTE: If Client RemoveIPAddressFilter request message does not contain "<IPv6Address>" tag then Test shall be deemed as "NOT DETECTED".

PASS -

• Client RemoveIPAddressFilter request messages are valid according to XML Schemas listed in Namespaces AND

• Client RemoveIPAddressFilter request in Test Procedure fulfills the following requirements:
  • [S1] Client request contains "<RemoveIPAddressFilter>" tag after the "<Body>" tag AND
  • [S3] "<RemoveIPAddressFilter>" includes tag: "<IPv6Address>" AND
  • [S4] "<IPv6Address>" includes tag: "<Address>" with specific IPv6 address value AND
  • [S5] "<IPv6Address>" includes tag: "<PrefixLength>" with value range from "0" to "128" AND
  • [S6] Device response contains "HTTP/* 200 OK" AND

FAIL -

• The Client failed PASS criteria.

6.4 Persistent Notification Storage Retrieval Test Cases

6.4.1 Feature Level Requirement:

Validated Feature: Persistent Notification Storage Retrieval (PersistentNotificationStorageRetrieval)

Check Condition based on Device Features: Persistent Notification Storage is supported by Device.

Required Number of Devices: 1
Profile C Requirement: Conditional

Profile A Requirement: Conditional

6.4.2 Expected Scenarios Under Test:

1. Client subscribes to device messages using CreatePullPointSubscription operation.

2. Client uses Seek method to change position of the pull pointer to include all NotificationMessages in the persistent storage with UtcTime attribute greater than or equal to the Seek argument.

3. Client uses Pull Point event mechanism to retrieve notification events from Device.

4. Client is considered as supporting Persistent Notification Storage Retrieval if the following conditions are met:
   • Client is able to seek stored events in Device using the Seek operation.

5. Client is considered as NOT supporting Persistent Notification Storage Retrieval if ANY of the following is TRUE:
   • No Valid Device Response to Seek request.

6.4.3 SEEK

Test Label: Persistent Notification Storage Retrieval - Seek

Test Case ID: PERSISTENTNOTIFICATIONSTORAGERETRIEVAL-1

Feature Under Test: Seek (PersistentNotificationStorageRetrieval_Seek)

Profile C Normative Reference: Conditional

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to seek stored events in Device using Pull Point event mechanism and Seek operation.

Pre-Requisite:

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

Test Procedure (expected to be reflected in network trace file):
1. Client invokes CreatePullPointSubscription message.

2. Device responds with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.

3. Client invokes Seek message to re-adjust the pull pointer into the past.

4. Device responds with code HTTP 200 OK and SeekResponse message.

5. Client invokes PullMessages command with Timeout and MessageLimit elements.


Test Result:

PASS -

• Client CreatePullPointSubscription request messages are valid according to XML Schemas listed in Namespaces AND

• Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:
  • [S1] Client request contains "<CreatePullPointSubscription>" tag after the "<Body>" tag AND
  • [S2] Device response contains "HTTP/* 200 OK" AND
  • [S3] Device response contains "<CreatePullPointSubscriptionResponse>" tag AND

• Client Seek request messages are valid according to XML Schemas listed in Namespaces AND

• Client Seek request in Test Procedure fulfills the following requirements:
  • [S4] Client request contains "<Seek>" tag after the "<Body>" tag AND
  • [S6] Device response contains "HTTP/* 200 OK" AND
  • [S7] Device response contains "<SeekResponse>" tag AND

• Client PullMessages request messages are valid according to XML Schemas listed in Namespaces AND

• Client PullMessages request in Test Procedure fulfills the following requirements:
  • [S8] Client request contains "<PullMessages>" tag after the "<Body>" tag AND
  • [S11] Device response contains "HTTP/* 200 OK" AND

FAIL -

• The Client failed PASS criteria.

6.5 System Date and Time Configuration Test Cases

6.5.1 Feature Level Requirement:

Validated Feature: System Date and Time Configuration (SystemDateAndTimeConfiguration)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Conditional

Profile C Requirement: Optional

Profile G Requirement: Optional

Profile Q Requirement: Conditional

Profile S Requirement: Optional

6.5.2 Expected Scenarios Under Test:

1. Client connects to Device to configure system date and time.

2. Client is considered as supporting System Date and Time Configuration if the following conditions are met:

   • Client is able to retrieve a system date and time using GetSystemDateAndTime operation AND

   • Client is able to configure a system date and time using EITHER SetSystemDateAndTime operation OR SetNTP operation.

3. Client is considered as NOT supporting System Date and Time Configuration if ANY of the following is TRUE:

   • No valid responses for GetSystemDateAndTime request OR

   • No valid responses for SetSystemDateAndTime request if detected AND
6.5.3 GET SYSTEM DATE AND TIME

Test Label: System Date and Time Configuration - Get System Date And Time

Test Case ID: SYSTEMDATEANDTIMECONFIGURATION-1

Feature Under Test: Get System Date And Time (SystemDateAndTimeConfiguration_GetSystemDateAndTime)

Profile A Normative Reference: Conditional

Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Conditional

Profile S Normative Reference: Optional

Test Purpose: To verify that Device system date and time is received by Client using the GetSystemDateAndTime operation.

Pre-Requisite:

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

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

1. Client invokes GetSystemDateAndTime request message to retrieve system date and time from the Device.
2. Device responds with code HTTP 200 OK and GetSystemDateAndTimeResponse message.

Test Result:

PASS -

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

- Client GetSystemDateAndTime request in Test Procedure fulfills the following requirements:
  - [S1] soapenv:Body element has child element tds:GetSystemDateAndTime AND
• Device response on the **GetSystemDateAndTime** request fulfills the following requirements:

  • [S2] It has HTTP 200 response code AND
  • [S3] soapenv:Body element has child element tds:GetSystemDateAndTimeResponse.

FAIL -

  • The Client failed PASS criteria.

### 6.5.4 SET SYSTEM DATE AND TIME

**Test Label:** System Date and Time Configuration - Set System Date And Time

**Test Case ID:** SYSTEMDATEANDTIMECONFIGURATION-2

**Feature Under Test:** Set System Date And Time

(SystemDateAndTimeConfiguration_SetSystemDateAndTime)

**Profile A Normative Reference:** Conditional

**Profile C Normative Reference:** Optional

**Profile G Normative Reference:** Optional

**Profile Q Normative Reference:** Conditional

**Profile S Normative Reference:** Optional

**Test Purpose:** To verify that Client is able to configure system date and time on Device using the **SetSystemDateAndTime** operation.

**Pre-Requisite:**

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

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

1. Client invokes **SetSystemDateAndTime** request message to set Device system date and time.

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

**Test Result:**
PASS -

- Client `SetSystemDateAndTime` request messages are valid according to XML Schemas listed in Namespaces AND

- Client `SetSystemDateAndTime` request in Test Procedure fulfills the following requirements:
  - [S1] `soapenv:Body` element has child element `tds:SetSystemDateAndTime` AND
  - [S2] If `tds:DateTimeType` element value is equal to "Manual" THEN `tds:SetSystemDateAndTime` contains `tds:UTCDatetime` element AND

- Device response on the `SetSystemDateAndTime` request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND

FAIL -

- The Client failed PASS criteria.

6.6 Get Access Profile Details Test Cases

6.6.1 Feature Level Normative Reference:

**Validated Feature:** Get Access Profile Details (GetAccessProfileDetails)

**Check Condition based on Device Features:** Access Rules Service is supported by Device.

**Required Number of Devices:** 1

**Profile A Requirement:** Conditional

6.6.2 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.

### 6.6.3 GET ACCESS PROFILES

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

**Test Case ID:** GETACCESSPROFILESDETAILS-1

**Feature Under Test:** Get Access Profiles (GetAccessProfileDetails_GetAccessProfiles)

**Profile A Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to get access profiles details 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 details 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.

6.7 Configure Access Profiles Test Cases

6.7.1 Feature Level Normative Reference:

Validated Feature: Configure Access Profiles (ConfigureAccessProfiles)

Check Condition based on Device Features: Access Rules Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.7.2 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 modifies 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.

### 6.7.3 CREATE ACCESS PROFILE

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

**Test Case ID:** CONFIGUREACCESSPROFILES-1

**Feature Under Test:** Create Access Profile (ConfigureAccessProfiles_CreateAccessProfile)

**Profile A Normative Reference:** Conditional

**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

FAIL -
• The Client failed PASS criteria.

6.7.4 MODIFY ACCESS PROFILE

Test Label: Configure Access Profiles - Modify Access Profile

Test Case ID: CONFIGUREACCESSPROFILES-2

Feature Under Test: Modify Access Profile (ConfigureAccessProfiles_ModifyAccessProfile)

Profile A Normative Reference: Conditional

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.

6.7.5 DELETE ACCESS PROFILE

Test Label: Configure Access Profiles - Delete Access Profile

Test Case ID: CONFIGUREACCESSPROFILES-3

Feature Under Test: Delete Access Profile (ConfigureAccessProfiles_DeleteAccessProfile)

Profile A Normative Reference: Conditional

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.

6.8 Get Credential State Test Cases

6.8.1 Feature Level Normative Reference:

Validated Feature: Get Credential State (GetCredentialState)

Check Condition based on Device Features: Credential Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.8.2 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.

6.8.3 GET CREDENTIAL STATE

Test Label: Get Credential State

Test Case ID: GETCREDENTIALSTATE-1

Feature Under Test: Get Credential State (GetCredentialState_GetCredentialStateRequest)

Profile A Normative Reference: Conditional

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.

6.9 Change Credential State Test Cases

6.9.1 Feature Level Normative Reference:

Validated Feature: Change Credential State (ChangeCredentialState)

Check Condition based on Device Features: Credential Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional
6.9.2 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.

6.9.3 ENABLE CREDENTIAL

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

**Test Case ID:** CHANGECREDENTIALSTATE-1

**Feature Under Test:** Enable Credential (ChangeCredentialState_EnableCredential)

**Profile A Normative Reference:** Conditional

**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

FAIL -

- The Client failed PASS criteria.

6.9.4 DISABLE CREDENTIAL

Test Label: Change Credential State - Disable Credential

Test Case ID: CHANGECREDENTIALSTATE-2

Feature Under Test: Disable Credential (ChangeCredentialState_DisableCredential)

Profile A Normative Reference: Conditional

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

FAIL -

- The Client failed PASS criteria.

6.10 Get Schedule Details Test Cases

6.10.1 Feature Level Normative Reference:

Validated Feature: Get Schedule Details (GetScheduleDetails)

Check Condition based on Device Features: Schedule Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.10.2 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.
6.10.3 GET SCHEDULES

Test Label: Get Schedule Details - Get Schedules

Test Case ID: GETSCHEDULEDETAILS-1

Feature Under Test: Get Schedules (GetScheduleDetails_GetSchedules)

Profile A Normative Reference: Conditional

Test Purpose: To verify that Client is able to get schedules details 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 details 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 -
6.11 Configure Schedules Test Cases

6.11.1 Feature Level Normative Reference:

Validated Feature: Configure Schedules (ConfigureSchedules)

Check Condition based on Device Features: Schedule Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.11.2 Expected Scenarios Under Test:

1. Client creates schedule on a Device using CreateSchedule operation.
2. Client modifies 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.

6.11.3 CREATE SCHEDULE

Test Label: Configure Schedules - Create Schedule

Test Case ID: CONFIGURESCHEDULES-1
Feature Under Test: Create Schedule (ConfigureSchedules_CreateSchedule)

Profile A Normative Reference: Conditional

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

FAIL -

- The Client failed PASS criteria.

6.11.4 MODIFY SCHEDULE

Test Label: Configure Schedules - Modify Schedule

Test Case ID: CONFIGURESCHEDULES-2
Feature Under Test: Modify Schedule (ConfigureSchedules_ModifySchedule)

Profile A Normative Reference: Conditional

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

FAIL -

- The Client failed PASS criteria.

6.11.5 DELETE SCHEDULE

Test Label: Configure Schedules - Delete Schedule

Test Case ID: CONFIGURESCHEDULES-3

Feature Under Test: Delete Schedule (ConfigureSchedules_DeleteSchedule)
Profile A Normative Reference: Conditional

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

FAIL -

- The Client failed PASS criteria.

6.12 Get Schedule State Test Cases

6.12.1 Feature Level Normative Reference:

Validated Feature: Get Schedule State (GetScheduleState)
Check Condition based on Device Features: Schedule Service is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.12.2 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_WSBaseNotification 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_WSBaseNotification feature when Client supports tns1:Schedule/State/Active notification if Device supports StateReporting.

6.12.3 GET SCHEDULE STATE

Test Label: Get Schedule Sate

Test Case ID: GETSCHEDULESTATE-1

Feature Under Test: Get Schedule State (GetScheduleState_GetScheduleStateRequest)

Profile A Normative Reference: Conditional

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.

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

FAIL -

- The Client failed PASS criteria.

6.13 Reset Antipassback Violation Test Cases

6.13.1 Feature Level Normative Reference:

Validated Feature: Reset Antipassback Violation (ResetAntipassbackViolation)

Check Condition based on Device Features: Reset Antipassback Violation is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional
6.13.2 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.

6.13.3 RESET ANTIPASSBACK VIOLATIONS

Test Label: Reset Antipassback Violation

Test Case ID: RESETANTIPASSBACKVIOLATION-1

Feature Under Test: Reset Antipassback Violation (ResetAntipassbackViolation_ResetAntipassbackViolationRequest)

Profile A Normative Reference: Conditional

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


FAIL -

- The Client failed PASS criteria.

6.14 Antipassback Violation Notifications Test Cases

6.14.1 Feature Level Normative Reference:

Validated Feature: Antipassback Violation Notifications (AntipassbackViolationNotifications)

Check Condition based on Device Features: Reset Antipassback Violation is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.14.2 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.

6.15 Get Special Day Group List Test Cases

6.15.1 Feature Level Normative Reference:

**Validated Feature:** Get Special Day Group List (GetSpecialDayGroupList)

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

**Required Number of Devices:** 1

**Profile A Requirement:** Conditional

6.15.2 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 received in GetSpecialDayGroupInfoList response in tsc:NextStartReference element OR

• GetSpecialDayGroupList request contains tsc:StartReference element value that was not received in GetSpecialDayGroupList response in tsc:NextStartReference element OR

• Complete Special Day Groups list was not received.

6.15.3 LISTING OF SPECIAL DAY GROUPS

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

Test Case ID: GETSPECIALDAYGROUPLIST-1

Feature Under Test: Get Special Day Group List
(GetSpecialDayGroupList_GetSpecialDayGroupListRequest)

Profile A Normative Reference: Optional

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.


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.

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.

6.15.4 LISTING OF SPECIAL DAY GROUP INFO

Test Label: Get Special Day Groups List - Listing of Special Day Group Info
Test Case ID: GETSPECIALDAYGROUPLIST-2

Feature Under Test: Get Special Day Group Info List
(GetSpecialDayGroupList_GetSpecialDayGroupInfoListRequest)

Profile A Normative Reference: Conditional

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.


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.


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.

### 6.16 Get Special Day Group Details Test Cases

#### 6.16.1 Feature Level Normative Reference:

**Validated Feature:** Get Special Day Group Details (GetSpecialDayGroupDetails)

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

**Required Number of Devices:** 1

**Profile A Requirement:** Conditional

#### 6.16.2 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 special day group listed in it if Device supports SpecialDays.

6.16.3 GET SPECIAL DAY GROUPS

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

**Test Case ID:** GETSPECIALDAYGROUPDETAILS-1

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

( GetSpecialDayGroupDetails_GetSpecialDayGroups )

**Profile A Normative Reference:** Conditional

**Test Purpose:** To verify that Client is able to get special day groups details 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 details 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.

6.17 Configure Special Day Groups Test Cases

6.17.1 Feature Level Normative Reference:

Validated Feature: Configure Special Day Groups (ConfigureSpecialDayGroups)

Check Condition based on Device Features: Special Days is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Conditional

6.17.2 Expected Scenarios Under Test:

1. Client creates special day group on a Device using CreateSpecialDayGroup operation.

2. Client modifies 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.

6.17.3 CREATE SPECIAL DAY GROUP

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

**Test Case ID:** CONFIGURESPECIALDAYGROUPS-1

**Feature Under Test:** Create Special Day Group
(ConfigureSpecialDayGroups_CreateSpecialDayGroup)

**Profile A Normative Reference:** Conditional

**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

FAIL -

• The Client failed PASS criteria.

6.17.4 MODIFY SPECIAL DAY GROUP

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

Test Case ID: CONFIGURESPECIALDAYGROUPS-2

Feature Under Test: Modify Special Day Group (ConfigureSpecialDayGroups_ModifySpecialDayGroup)

Profile A Normative Reference: Conditional

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

FAIL -

- The Client failed PASS criteria.

6.17.5 DELETE SPECIAL DAY GROUP

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

Test Case ID: CONFIGURESPECIALDAYGROUPS-3

Feature Under Test: Delete Special Day Group
(ConfigureSpecialDayGroups_DeleteSpecialDayGroup)

Profile A Normative Reference: Conditional

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 \texttt{DeleteSpecialDayGroupResponse} message.

\textbf{Test Result:}

\textbf{PASS -}

- Client \texttt{DeleteSpecialDayGroup} request messages are valid according to XML Schemas listed in \texttt{Namespaces} AND

- Client \texttt{DeleteSpecialDayGroup} request in Test Procedure fulfills the following requirements:
  - \texttt{[S1] soapenv:Body} element has child element \texttt{tsc:DeleteSpecialDayGroup} AND

- Device response on the \texttt{DeleteSpecialDayGroup} request fulfills the following requirements:
  - \texttt{[S2]} It has HTTP 200 response code AND
  - \texttt{[S3] \texttt{soapenv:Body} element has child element \texttt{tsc:DeleteSpecialDayGroupResponse}}.

\textbf{FAIL -}

- The Client failed PASS criteria.

\subsection*{6.18 Special Days Notifications Test Cases}

\subsubsection*{6.18.1 Feature Level Normative Reference:}

\textbf{Validated Feature:} Special Days Notifications (SpecialDaysNotifications)

\textbf{Check Condition based on Device Features:} Special Days is supported by Device.

\textbf{Required Number of Devices:} 1

\textbf{Profile A Requirement:} Conditional

\subsubsection*{6.18.2 Expected Scenarios Under Test:}

1. Client subscribes to device messages using \texttt{CreatePullPointSubscription} operation OR \texttt{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.
7 Test Cases for Profile Optional Features

7.1 Get Services with Capabilities Test Cases

7.1.1 Feature Level Requirement:

Validated Feature: Get Services with Capabilities (GetServicesWithCapabilities)

Check Condition based on Device Features: GetServices is supported by Device.

Required Number of Devices: 1

Profile A Requirement: Optional
Profile C Requirement: Optional
Profile D Requirement: Optional
Profile G Requirement: Optional
Profile Q Requirement: Optional

7.1.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a service capabilities.

2. Client is considered as supporting Get Services with Capabilities if the following conditions are met:
   • Client is able to retrieve a services capabilities using GetServices operation.

3. Client is considered as NOT supporting Get Services with Capabilities if ANY of the following is TRUE:
   • No valid responses for GetServices request.

7.1.3 GET SERVICES

Test Label: Get Services with Capabilities - Get Services

Test Case ID: GETSERVICESWITHCAPABILITIES-1

Feature Under Test: Get Services with Capabilities
(GetServicesWithCapabilities_GetServicesWithCapabilitiesRequest)
Profile A Normative Reference: Optional

Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Optional

Profile D Normative Reference: Optional

Test Purpose: To verify that services capabilities provided by Device is received by Client using the GetServices operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetServices operation with tds:IncludeCapability element equal to true present.
- The Device supports GetServices command.

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

1. Client invokes GetServices request message with tds:IncludeCapability element equal to true to retrieve redential service capabilities from the Device.

2. Device responds with code HTTP 200 OK and GetServicesResponse message.

Test Result:

PASS -

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

- Client GetServices request in Test Procedure fulfills the following requirements:
  - [S1] soapenv:Body element has child element tds:GetServices AND
  - [S2] It contains tds:IncludeCapability element equal to true AND

- Device response on the GetServices request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND

FAIL -
7.2 Set Synchronization Point Test Cases

7.2.1 Feature Level Requirement:

Validated Feature: Set Synchronization Point (SetSynchronizationPoint)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile S Requirement: Optional

Profile Q Requirement: Optional

Profile G Requirement: Optional

Profile T Requirement: Mandatory

Profile D Requirement: Mandatory

7.2.2 Expected Scenarios Under Test:

1. Client connects to Device to synchronize property states.

2. Client is considered as supporting Set Synchronization Point if the following conditions are met:
   - Client is able to synchronize property states using SetSynchronizationPoint operation for subscriptions AND

3. Client is considered as NOT supporting Set Synchronization Point if the following is TRUE:
   - No valid responses for SetSynchronizationPoint request OR
   - SetSynchronizationPoint request does not contains valid wsa:Action header.

7.2.3 SET SYNCHRONIZATION POINT

Test Label: Set Synchronization Point - Set Synchronization Point
Test Case ID: SETSYNCHRONIZATIONPOINT-1

Feature Under Test: Set Synchronization Point
(SetSynchronizationPoint_SetSynchronizationPointAction)

Profile A Normative Reference: Mandatory
Profile C Normative Reference: Mandatory
Profile S Normative Reference: Conditional
Profile Q Normative Reference: Optional
Profile G Normative Reference: Conditional
Profile T Normative Reference: Mandatory
Profile D Normative Reference: Mandatory

Test Purpose: To verify that the Client is able to use SetSynchronizationPoint operation for subscription.

Pre-Requisite:

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

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

1. Client invokes SetSynchronizationPoint message with valid wsa:Action header to synchronize its properties with the properties of the device.
2. Device responses with code HTTP 200 OK and SetSynchronizationPointResponse message.

Test Result:

PASS -

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

- Client SetSynchronizationPoint request in Test Procedure fulfills the following requirements:
  - [S1] soapenv:Body element has child element tev:SetSynchronizationPoint AND
  - [S2] It contains wsa:Action element in header equal to "http://www.onvif.org/ver10/events/wsdl/PullPointSubscription/SetSynchronizationPointRequest" AND

- Device response on the SetSynchronizationPoint request fulfills the following requirements:
• [S3] It has HTTP 200 response code AND
• [S4] soapenv:Body element has child element tev:SetSynchronizationPointResponse

FAIL -

• The Client failed PASS criteria.

7.3 Unsubscribe Test Cases

Validated Feature: Unsubscribe (Unsubscribe)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile S Requirement: Optional

Profile Q Requirement: Optional

Profile G Requirement: Optional

Profile T Requirement: Optional

7.3.1 Expected Scenarios Under Test:

1. Client connects to Device to Unsubscribe subscriptions.

2. Client is considered as supporting Unsubscribe if the following conditions are met:
   • Client is able to unsubscribe subscriptions using Unsubscribe operation.

3. Client is considered as NOT supporting Unsubscribe if the following is TRUE:
   • No valid responses for Unsubscribe request OR
   • Unsubscribe request does not contains valid wsa:Action header.

7.3.2 UNSUBSCRIBE

Test Label: Unsubscribe - Unsubscribe

Test Case ID: UNSUBSCRIBE-1
Feature Under Test: Unsubscribe (Unsubscribe_UnsubscribeAction)

Profile A Normative Reference: Mandatory
Profile C Normative Reference: Mandatory
Profile S Normative Reference: Conditional
Profile Q Normative Reference: Optional
Profile G Normative Reference: Conditional
Profile T Normative Reference: Optional

Test Purpose: To verify that the Client is able to use Unsubscribe operation to terminate a subscription.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with Unsubscribe operations present.

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

1. Client invokes Unsubscribe message with valid wsa:Action header to terminate a subscription.

2. Device responses with code HTTP 200 OK and UnsubscribeResponse message.

Test Result:

PASS -

• Client Unsubscribe request messages are valid according to XML Schemas listed in Namespaces AND

• Client Unsubscribe request in Test Procedure fulfills the following requirements:
  • [S1] soapenv:Body element has child element wsnt:Unsubscribe AND
  • [S2] It contains wsa:Action element in header equal to "http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest" AND

• Device response on the Unsubscribe request fulfills the following requirements:
  • [S3] It has HTTP 200 response code AND
  • [S4] soapenv:Body element has child element wsnt:UnsubscribeResponse

FAIL -

• The Client failed PASS criteria.
7.4 Hostname Configuration Test Cases

7.4.1 Feature Level Requirement:

Validated Feature: Hostname Configuration (HostnameConfiguration)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile G Requirement: Optional

Profile Q Requirement: Conditional

Profile S Requirement: Optional

7.4.2 Expected Scenarios Under Test:

1. Client connects to Device to configure hostname.

2. Client is considered as supporting Hostname Configuration if the following conditions are met:
   • Client is able to retrieve a hostname information from the Device using `GetHostname` operation AND
   • Client is able set a network hostname on the Device using `SetHostname` operation.

3. Client is considered as NOT supporting Hostname Configuration if ANY of the following is TRUE:
   • No valid responses for `GetHostname` request OR
   • No valid responses for `SetHostname` request.

7.4.3 GET HOSTNAME

Test Label: Hostname Configuration - Get Hostname

Test Case ID: HOSTNAMECONFIGURATION-1

Feature Under Test: Get Hostname (HostnameConfiguration_GetHostname)
Profile A Normative Reference: Optional

Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Conditional

Profile S Normative Reference: Optional

Test Purpose: To verify that hostname settings of the Device are received by Client using the GetHostname operation.

Pre-Requisite:

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

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

1. Client invokes GetHostname request message to retrieve hostname from the Device.
2. Device responds with code HTTP 200 OK and GetHostnameResponse message.

Test Result:

PASS -

• Client GetHostname request messages are valid according to XML Schemas listed in Namespaces AND

• Client GetHostname request in Test Procedure fulfills the following requirements:
  
  • [S1] soapenv:Body element has child element tds:GetHostname AND

• Device response on the GetHostname request fulfills the following requirements:
  
  • [S2] It has HTTP 200 response code AND
  
  • [S3] soapenv:Body element has child element tds:GetHostnameResponse.

FAIL -

• The Client failed PASS criteria.

7.4.4 SET HOSTNAME

Test Label: Hostname Configuration - Set Hostname
Test Case ID: HOSTNAMECONFIGURATION-2

Feature Under Test: Set Hostname (HostnameConfiguration_SetHostname)

Profile A Normative Reference: Optional

Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Conditional

Profile S Normative Reference: Optional

Test Purpose: To verify that Client is able to set the Hostname settings on Device using the SetHostname operation.

Pre-Requisite:

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

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

1. Client invokes SetHostname request message to set hostname on the Device.

2. Device responds with code HTTP 200 OK and SetHostnameResponse message.

Test Result:

PASS -

1. Client SetHostname request messages are valid according to XML Schemas listed in Namespaces AND

2. Client SetHostname request in Test Procedure fulfills the following requirements:

   • [S1] soapenv:Body element has child element tds:SetHostname AND

3. Device response on the SetHostname request fulfills the following requirements:

   • [S2] It has HTTP 200 response code AND

   • [S3] soapenv:Body element has child element tds:SetHostnameResponse.

FAIL -

1. The Client failed PASS criteria.
7.5 DNS Configuration Test Cases

7.5.1 Feature Level Requirement:

Validated Feature: DNS Configuration (DNSConfiguration)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile G Requirement: Optional

Profile Q Requirement: Conditional

Profile S Requirement: Optional

7.5.2 Expected Scenarios Under Test:

1. Client connects to Device to configure a domain name server.

2. Client is considered as supporting DNS Configuration if the following conditions are met:
   - Client is able to get DNS settings from the Device using GetDNS operation AND
   - Client is able to set DNS settings on the Device using SetDNS operation.

3. Client is considered as NOT supporting DNS Configuration if ANY of the following is TRUE:
   - No valid responses for GetDNS request OR
   - No valid responses for SetDNS request.

7.5.3 GET DNS

Test Label: DNS Configuration - Get DNS

Test Case ID: DNSCONFIGURATION-1

Feature Under Test: Get DNS (DNSConfiguration_GetDNS)

Profile A Normative Reference: Optional
Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Conditional

Profile S Normative Reference: Optional

Test Purpose: To verify that DNS settings of Device are received by Client using the GetDNS operation.

Pre-Requisite:

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

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

1. Client invokes GetDNS request message to retrieve DNS settings from the Device.
2. Device responds with code HTTP 200 OK and GetDNSResponse message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

7.5.4 SET DNS

Test Label: DNS Configuration - Set DNS
Test Case ID: DNSCONFIGURATION-2

Feature Under Test: Set DNS (DNSConfiguration_SetDNS)

Profile A Normative Reference: Optional
Profile C Normative Reference: Optional
Profile G Normative Reference: Optional
Profile Q Normative Reference: Conditional
Profile S Normative Reference: Optional

Test Purpose: To verify that Client is able to set the DNS settings on Device using the SetDNS operation.

Pre-Requisite:

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

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

1. Client invokes SetDNS request message to set hostname on the Device.
2. Device responds with code HTTP 200 OK and SetDNSResponse message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.
7.6 Network Protocols Configuration Test Cases

7.6.1 Feature Level Requirement:

Validated Feature: Network Protocols Configuration (NetworkProtocolsConfiguration)

Check Condition based on Device Features: None

Required Number of Devices: 1

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile G Requirement: Optional

Profile Q Requirement: Conditional

Profile S Requirement: Optional

7.6.2 Expected Scenarios Under Test:

1. Client connects to Device to configure a network protocols.

2. Client is considered as supporting Network Protocols Configuration if the following conditions are met:
   • Client is able to get defined network protocols from the Device using GetNetworkProtocols operation AND
   • Client is able configures defined network protocols on the Device using SetNetworkProtocols operation.

3. Client is considered as NOT supporting Network Protocols Configuration if ANY of the following is TRUE:
   • No valid responses for GetNetworkProtocols request OR
   • No valid responses for SetNetworkProtocols request.

7.6.3 GET NETWORK PROTOCOLS

Test Label: Network Protocols Configuration - Get Network Protocols

Test Case ID: NETWORKPROTOCOLSCONFIGURATION-1
Feature Under Test: Get Network Protocols
(NetworkProtocolsConfiguration_GetNetworkProtocols)

Profile A Normative Reference: Optional
Profile C Normative Reference: Optional
Profile G Normative Reference: Optional
Profile Q Normative Reference: Conditional
Profile S Normative Reference: Optional

Test Purpose: To verify that network protocols of Device are received by Client using the GetNetworkProtocols operation.

Pre-Requisite:

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

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

1. Client invokes GetNetworkProtocols request message to retrieve network protocols from the Device.

Test Result:

PASS -

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

- Client GetNetworkProtocols request in Test Procedure fulfills the following requirements:
  - [S1] soapenv:Body element has child element tds:GetNetworkProtocols AND

- Device response on the GetNetworkProtocols request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND

FAIL -

- The Client failed PASS criteria.
7.6.4 SET NETWORK PROTOCOLS

Test Label: Network Protocols Configuration - Set Network Protocols

Test Case ID: NETWORKPROTOCOLSCONFIGURATION-2

Feature Under Test: Set Network Protocols
(NetworkProtocolsConfiguration_SetNetworkProtocols)

Profile A Normative Reference: Optional

Profile C Normative Reference: Optional

Profile G Normative Reference: Optional

Profile Q Normative Reference: Conditional

Profile S Normative Reference: Optional

Test Purpose: To verify that Client is able to configure defined network protocols on Device using the SetNetworkProtocols operation.

Pre-Requisite:

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

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


Test Result:

PASS -

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

FAIL -

• The Client failed PASS criteria.
8 Supplementary Features and Test Cases

8.1 METADATA STREAMING USING MEDIA2

Test Label: Metadata Streaming Using Media2

Test Case ID: MEDIA2_METADATASTREAMING-1

Feature Under Test: Metadata Streaming

Profile T Normative Reference: Conditional

Profile M Normative Reference: Mandatory

Test Purpose: To verify that the Client is able to retrieve the Metadata Streaming.

Pre-Requisite:

• The Network Trace Capture files contains at least one Conversation between Client and Device with Metadata Streaming using Media2 Service.

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

1. Client invokes GetStreamUri request message for Media2 service for media profile that contains Metadata Configuration. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.

2. Device responds with code HTTP 200 OK and GetStreamUriResponse message.

3. Client invokes RTSP DESCRIBE request to retrieve media stream description.

4. Device responds with code RTSP 200 OK and SDP information with Media Type: "application" and with encoding name "vnd.onvif.metadata" or "vnd.onvif.metadata.gzip" or "vnd.onvif.metadata.exi.onvif" or "vnd.onvif.metadata.exi.ext".

5. Client invokes RTSP SETUP request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for metadata streaming.

6. Device responds with code RTSP 200 OK.

7. Client invokes RTSP PLAY request without "onvif-replay" Require header to start media stream.

8. Device responds with code RTSP 200 OK.

9. Client invokes RTSP TEARDOWN request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

**Test Result:**

**Note:** RTSP requests and RTSP response could be tunneled in HTTP if RtspOverHttp transport is used.

**PASS -**

- There is Client RTSP DESCRIBE request in Test Procedure
- Device response on the RTSP DESCRIBE request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "application" (m=application) with sessions attribute "rtpmap" with encoding name "vnd.onvif.metadata" OR "vnd.onvif.metadata.gzip" OR "vnd.onvif.metadata.exi.onvif" OR "vnd.onvif.metadata.exi.ext" (see ONVIF Streaming Spec) AND
- There is Client RTSP SETUP request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client RTSP DESCRIBE request AND
  - [S4] It invoked after the Client RTSP DESCRIBE request AND
  - [S5] RTSP address that was used to send RTSP SETUP is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain Require request header field with value is equal to "onvif-replay" AND
- Device response on the RTSP SETUP request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the GetStreamUri request invoked for Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client RTSP DESCRIBE request AND
  - [S10] It received before the Client RTSP DESCRIBE request AND
  - [S11] It contains tr2:GetStreamUriResponse\tr2:Uri element which value is equal to RTSP address that was used to send the RTSP DESCRIBE request AND
There is Client RTSP PLAY request in Test Procedure fulfills the following requirements:

- [S12] It invoked for the same Device as for the Client RTSP SETUP request AND
- [S13] It invoked after the Client RTSP SETUP request AND
- [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S15] It does not contain Require request header field with value is equal to "onvif-replay" AND

Device response on the RTSP PLAY request fulfills the following requirements:

- [S16] It has RTSP 200 response code AND

There is Client RTSP TEARDOWN request in Test Procedure fulfills the following requirements:

- [S17] It invoked for the same Device as for the Client RTSP SETUP request AND
- [S18] It invoked after the Client RTSP PLAY request AND
- [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND

If there is Device response on the RTSP TEARDOWN request then it fulfills the following requirements:

- [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.
Annex A Test for Appendix A

A.1 Required Number of Devices Summary

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

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

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Feature Name</th>
<th>Required Number of Devices</th>
<th>Check Condition based on Device Features</th>
<th>Check Condition based on Device Features ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>tc.HTTPDigest</td>
<td>HTTP Digest</td>
<td>3</td>
<td>Digest</td>
<td>Digest</td>
</tr>
<tr>
<td>tc.Capabilities</td>
<td>Capabilities</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.GetServices</td>
<td>Get Services</td>
<td>3</td>
<td>GetServices is supported by Device.</td>
<td>GetServices</td>
</tr>
<tr>
<td>tc.EventHandling</td>
<td>Event Handling</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.KeepAliveForPullPointEventHandling</td>
<td>Keep Alive for Pull Point Event Handling</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.Discovery</td>
<td>Discovery</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.DeviceDiscoveryTypeFilter</td>
<td>Device Discovery Type Filter</td>
<td>3</td>
<td>Device Discovery Type is supported by Device.</td>
<td>DiscoveryTypesTdsDevice</td>
</tr>
<tr>
<td>tc.UserHandling</td>
<td>User Handling</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.GetCredentialCapabilities</td>
<td>Get Credential Capabilities</td>
<td>3</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.GetCredentialList</td>
<td>Get Credential List</td>
<td>3</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.GetCredentialDetails</td>
<td>Get Credential Details</td>
<td>3</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.ConfigureCredentials</td>
<td>Configure Credentials</td>
<td>3</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>Feature ID</td>
<td>Feature Name</td>
<td>Required Number of Devices</td>
<td>Check Condition based on Device Features</td>
<td>Check Condition based on Device Features ID</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>---------------------------</td>
<td>------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>tc.CredentialsNotifications</td>
<td>Credential Configuration and State Notifications</td>
<td>3</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.GetScheduleList</td>
<td>Get Schedule List</td>
<td>3</td>
<td>Schedule Service is supported by Device.</td>
<td>Schedule</td>
</tr>
<tr>
<td>tc.SchedulesNotifications</td>
<td>Schedule Configuration Notifications</td>
<td>3</td>
<td>Schedule Service is supported by Device.</td>
<td>Schedule</td>
</tr>
<tr>
<td>tc.GetAccessProfileList</td>
<td>Get Access Profile List</td>
<td>3</td>
<td>Access Rules Service is supported by Device.</td>
<td>AccessRulesService</td>
</tr>
<tr>
<td>tc.AccessProfileNotifications</td>
<td>Access Profile Configuration Notifications</td>
<td>3</td>
<td>Access Rules Service is supported by Device.</td>
<td>AccessRulesService</td>
</tr>
<tr>
<td>tc.NetworkConfiguration</td>
<td>Network Configuration</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.System</td>
<td>System</td>
<td>3</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.IPAddressFiltering</td>
<td>IP Address Filtering</td>
<td>1</td>
<td>IP Filter is supported by Device.</td>
<td>IPFilter</td>
</tr>
<tr>
<td>tc.PersistentNotificationStorageRetrieval</td>
<td>Persistent Notification Storage Retrieval</td>
<td>1</td>
<td>Persistent Notification Storage is supported by Device.</td>
<td>PersistentNotificationStorage</td>
</tr>
<tr>
<td>tc.SystemDateTimeConfiguration</td>
<td>System Date and Time Configuration</td>
<td>1</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.GetAccessProfileDetails</td>
<td>Get Access Profile Details</td>
<td>1</td>
<td>Access Rules Service is supported by Device.</td>
<td>AccessRulesService</td>
</tr>
<tr>
<td>Feature ID</td>
<td>Feature Name</td>
<td>Required Number of Devices</td>
<td>Check Condition based on Device Features</td>
<td>Check Condition based on Device Features ID</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>tc.Configure</td>
<td>Configure Access Profiles</td>
<td>1</td>
<td>Access Rules Service is supported by Device.</td>
<td>AccessRulesService</td>
</tr>
<tr>
<td>tc.GetCredentialState</td>
<td>Get Credential State</td>
<td>1</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.ChangeCredentialState</td>
<td>Change Credential State</td>
<td>1</td>
<td>Credential Service is supported by Device.</td>
<td>Credential</td>
</tr>
<tr>
<td>tc.GetScheduleDetails</td>
<td>Get Schedule Details</td>
<td>1</td>
<td>Schedule Service is supported by Device.</td>
<td>Schedule</td>
</tr>
<tr>
<td>tc.Configure Schedules</td>
<td>Configure Schedules</td>
<td>1</td>
<td>Schedule Service is supported by Device.</td>
<td>Schedule</td>
</tr>
<tr>
<td>tc.GetScheduleState</td>
<td>Get Schedule State</td>
<td>1</td>
<td>Schedule Service is supported by Device.</td>
<td>Schedule</td>
</tr>
<tr>
<td>tc.ResetAntipassbackViolation</td>
<td>Reset Antipassback Violation</td>
<td>1</td>
<td>Reset Antipassback Violation is supported by Device.</td>
<td>ResetAntipassbackViolation</td>
</tr>
<tr>
<td>tc.AntipassbackViolationNotifications</td>
<td>Antipassback Violation Notifications</td>
<td>1</td>
<td>Reset Antipassback Violation is supported by Device.</td>
<td>ResetAntipassbackViolation</td>
</tr>
<tr>
<td>tc.GetSpecialDayGroupList</td>
<td>Get Special Day Group List</td>
<td>1</td>
<td>Special Days is supported by Device.</td>
<td>SpecialDays</td>
</tr>
<tr>
<td>tc.GetSpecialDayGroupDetails</td>
<td>Get Special Day Group Details</td>
<td>1</td>
<td>Special Days is supported by Device.</td>
<td>SpecialDays</td>
</tr>
<tr>
<td>Feature ID</td>
<td>Feature Name</td>
<td>Required Number of Devices</td>
<td>Check Condition based on Device Features</td>
<td>Check Condition based on Device Features ID</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>tc.ConfigureSpecialDayGroups</td>
<td>Configure Special Day Groups</td>
<td>1</td>
<td>Special Days is supported by Device.</td>
<td>SpecialDays</td>
</tr>
<tr>
<td>tc.SpecialDaysNotifications</td>
<td>Special Days Notifications</td>
<td>1</td>
<td>Special Days is supported by Device.</td>
<td>SpecialDays</td>
</tr>
<tr>
<td>tc.GetServicesWithCapabilities</td>
<td>Get Services with Capabilities</td>
<td>1</td>
<td>GetServices is supported by Device.</td>
<td>GetServices</td>
</tr>
<tr>
<td>tc.SetSynchronizationPoint</td>
<td>Set Synchronization Point</td>
<td>1</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.HostnameConfiguration</td>
<td>Hostname Configuration</td>
<td>1</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.DNSConfiguration</td>
<td>DNS Configuration</td>
<td>1</td>
<td>None</td>
<td>All</td>
</tr>
<tr>
<td>tc.NetworkProtocolsConfiguration</td>
<td>Network Protocols Configuration</td>
<td>1</td>
<td>None</td>
<td>All</td>
</tr>
</tbody>
</table>