ONVIF Core Client Test Specification

Version 16.01 January, 2016

Table of Contents

Terms and Definitions

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].

Definitions

This section describes terms and definitions used in this document.

Profile See ONVIF Profile Policy.

Profile A The Profile A Specification.

ONVIF Device	Computer appliance or software program that exposes one or multiple ONVIF Web Services.
ONVIF Client	Computer appliance or software program that uses ONVIF Web Services.
Conversation	A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.
Network	A network is an interconnected group of devices communicating using the Internet protocol.
Network Trace Capture file	Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.
SOAP	SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.
Client Test Tool	ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.
Access Policy	An association of an access point and a schedule. An access policy defines when an access point can be accessed using an access profile which contains this access policy.
Access Profile	A collection of access policies, used to define role based access.
Access Point	A logical composition of a physical door and ID point(s) controlling access in one direction.

Credential	A physical/tangible object, a piece of knowledge, or a facet of a person's physical being, that enables an individual access to a given physical facility or computer-based information system.
Validity Period	From a certain point in time, to a later point in time.
Schedule	A set of time periods, for example: working hours (weekdays from 08:00 AM to 06:00 PM). It may also include one or more special days schedule.
ID Point	A device that converts reader signals to protocols recognized by an authorization engine. It can be card reader, REX, biometric reader etc.
Anti-Passback	Operating mode which requires user validation when leaving a security controlled area in order to be able to re-enter and vice versa.
Anti-Passback Violation State	A signal stating if the anti-passback rules have been violated for a credential.
Credential Format	The credential data can be formatted in many different ways. ONVIF supports the BACnet format types in [ISO 16484-5:2014-09 Annex P].
Credential Holder	Associates a credential with a user. Typically it holds a reference to a credential and a reference to a user.
Credential Identifier	Card number, unique card information, PIN, fingerprint, or other biometric information, etc., that can be validated in an access point.
Credential Number	A sequence of bytes uniquely identifying a credential at an access point.
Credential State	The credential state indicates if a credential is enabled or disabled. The state also indicates if anti-passback has been violated or not. The state may also contain a reason why the credential was disabled.
Duress	Forcing a person to provide access to a secure area against that person's wishes.
Format Type	See Credential Format.
iCalendar	An industry standard format for exchanging scheduling and activity-recording information electronically.
Special Days	A set of dates that require the regular Schedule to be overridden, e.g. holidays, half-days or working Sundays.
Special Days Schedule	A schedule that defines time periods for a Special Day List.
Time Period	A time period has a start time and an end time, e.g. 8 AM to 6 PM.
vEvent	A component in iCalendar, specifying the properties of an event.
Valid Device Response	Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.

Abbreviations

This section describes abbreviations used in this document.

PACS Physical Access Control System.

HTTP Hyper Text Transport Protocol.

HTTPS Hyper Text Transport Protocol over Secure Socket Layer.

URI Uniform Resource Identifier.

WSDL Web Services Description Language.

XML eXtensible Markup Language.

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 1. Defined namespaces in this specification

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

Advanced Pull Point Event Handling Test Cases

Validated Feature: advanced_pp_event_handling

Profile A Requirement: Optional

Profile C Requirement: Optional

Profile S Requirement: Optional

Profile Q Requirement: Optional

Profile G Requirement: Optional

Expected Scenarios Under Test:

1. Client connects to Device to initiate Pull Point Event Handling.
2. Client is considered as supporting Advanced Pull Point Event Handling if the following conditions are met:
   - Client is able to synchronize property states using **SetSynchronizationPoint** operation for pull point subscribtions AND
   - Client is able to unsubscribe pull point subscribtions using **Unsubscribe** operation.
3. Client is considered as NOT supporting Advanced Pull Point Event Handling if the following is TRUE:
   - Client supports EventHandling_Pullpoint feature AND
   - No valid responses for **SetSynchronizationPoint** request OR
   - **SetSynchronizationPoint** request does not contains valid **wsa:Action** header OR
   - **SetSynchronizationPoint** request contains valid reference parameters IF reference parameters was specified in **tev:SubscriptionReference\wsa:ReferenceParameters** element of corresponding **CreatePullPointSubscriptionResponse** message OR
   - **SetSynchronizationPoint** request was invoked to valid address which was specified in **tev:SubscriptionReference\wsa:Address** element of corresponding **CreatePullPointSubscriptionResponse** message OR
   - No valid responses for **Unsubscribe** request OR
   - **Unsubscribe** request does not contains valid **wsa:Action** header OR
   - **Unsubscribe** request contains valid reference parameters IF reference parameters was specified in **tev:SubscriptionReference\wsa:ReferenceParameters** element of corresponding **CreatePullPointSubscriptionResponse** message OR
   - **Unsubscribe** request was invoked to valif address which was specified in **tev:SubscriptionReference\wsa:Address** element of corresponding **CreatePullPointSubscriptionResponse** message

SET SYNCHRONIZATION POINT

Test Label: Advanced Pull Point Event Handling - Set Synchronization Point

Test Case ID: ADVANCEDPPEVENTHANDLING-3

Profile A Requirement: Mandatory

Profile C Requirement: Mandatory

Profile S Requirement: Conditional

Profile Q Requirement: Optional

Profile G Requirement: Conditional

Feature Under Test: SetSynchronizationPoint

Test Purpose: To verify that the Client is able to use **SetSynchronizationPoint** operation for Pull Point subscribtion.

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 **CreatePullPointSubscription** message.
2. Device responses with code HTTP 200 OK and **CreatePullPointSubscriptionResponse** message.
3. Client invokes **SetSynchronizationPoint** message to valid address recieved in **CreatePullPointSubscriptionResponse** message for the created Pull Point subscribtion with valid **wsa:Action** header and valid reference parameters recieved in **CreatePullPointSubscriptionResponse** message.
4. 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 **wsnt:SetSynchronizationPoint** AND
  - [S2] It contains **wsa:Action** element in header equal to "http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/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 **wsnt:SetSynchronizationPointResponse**
- There is a Device response on the **CreatePullPointSubscription** request in Test Procedure fulfills the following requirements:

- [S5] It has HTTP 200 response code AND
- [S6] It received for the same Device as for the Client **SetSynchronizationPoint** request AND
- [S7] It received before the Client **SetSynchronizationPoint** request AND
- [S8] It contains **tev:SubscriptionReference\wsa:Address** element which is equal to HTTP address that was used to send the **SetSynchronizationPoint** request AND
- [S9] All reference parameters listed in **tev:SubscriptionReference\wsa:ReferenceParameters** element was send in **SetSynchronizationPoint** request header with the same values and namespaces as in **CreatePullPointSubscriptionResponse** message and with **@wsa:IsReferenceParameter** attribute equal to true

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_pp_event_handling.set_synchronization_point

UNSUBSCRIBE

Test Label: Advanced Pull Point Event Handling - Unsubscribe

Test Case ID: ADVANCEDPPEVENTHANDLING-4

Profile A Requirement: Mandatory

Profile C Requirement: Mandatory

Profile S Requirement: Conditional

Profile Q Requirement: Optional

Profile G Requirement: Conditional

Feature Under Test: Unsubscribe

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

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 **CreatePullPointSubscription** message.
2. Device responses with code HTTP 200 OK and **CreatePullPointSubscriptionResponse** message.
3. Client invokes **Unsubscribe** message to valid address recieved in **CreatePullPointSubscriptionResponse** message for the created Pull Point subscribtion with valid **wsa:Action** header and valid reference parameters recieved in **CreatePullPointSubscriptionResponse** message.
4. 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**
- There is a Device response on the **CreatePullPointSubscription** request in Test Procedure fulfills the following requirements:
  - [S5] It has HTTP 200 response code AND
  - [S6] It received for the same Device as for the Client **Unsubscribe** request AND
  - [S7] It received before the Client **Unsubscribe** request AND
  - [S8] It contains **tev:SubscriptionReference\wsa:Address** element which is equal to HTTP address that was used to send the **Unsubscribe** request AND
  - [S9] All reference parameters listed in **tev:SubscriptionReference\wsa:ReferenceParameters** element was send in **Unsubscribe** request header with the same values and namespaces as in **CreatePullPointSubscriptionResponse** message and with **@wsa:IsReferenceParameter** attribute equal to true

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_pp_event_handling.unsubscribe

Recording Control – Dynamic Recording Configurations Test Cases

Expected Scenarios Under Test:

1. Client subscribes to device messages using CreatePullPointSubscription operation to get notifications about created and deleted recordings.
2. Client creates new Recording.
3. Client uses Pull Point event mechanism to retrieve notification events from Device.
4. Client delete Recording.
5. Client uses Pull Point event mechanism to retrieve notification events from Device.
6. Client is considered as supporting Dynamic Recording Configurations if the following conditions are met:
   - When Device supports DynamicRecordings feature and Client supports Dynamic Recordings Configurations capability:
     - Client is able to create a recording using the CreateRecording operation AND

- Client is able to delete a recording using the DeleteRecording operation AND
- Client supports EventHandling_Pullpoint feature AND
- Client is able to retrieve notifications about created recordings AND
- Client is able to retrieve notifications about deleted recordings.

7. Client is considered as NOT supporting Dynamic Recording Configurations if ANY of the following is TRUE:
   - When Device supports DynamicRecordings feature and Client supports Dynamic Recordings Configurations capability:
     - No Valid Device Response to CreateRecording request (except soap fault: MaxRecordings) OR
     - No Valid Device Response to DeleteRecording request (except soap fault: CannotDelete) OR
     - Client does not support EventHandling_Pullpoint feature OR
     - Client unable to retrieve notifications about created recordings OR
     - Client unable to retrieve notifications about deleted recordings OR

RETRIEVE CREATE RECORDING NOTIFICATION

Test Label: Dynamic Recording Configurations - Retrieve Notifications about created recordings

Test Case ID: DYNAMICRECORDINGCONFIGURATION-3

Profile G Requirement: Conditional

Feature Under Test: Dynamic Recording Configurations

Test Purpose: To verify that Client is able to retrieve notifications about created recordings from Device using the Pull Point event mechanism.

Pre-Requisite:

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

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

1. Client invokes CreatePullPointSubscription message to retrieve notification about created recording.
2. Device responses with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.

Test Result:

PASS -

- Client CreatePullPointSubscription request messages are valid according to XML Schemas listed in NamespacesAND
- Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:

- [S1] <soapenv:Body> element has child element <tev:CreatePullPointSubscription> AND
- [S2] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete THEN <wsnt:TopicExpression> element is equal to tns1:RecordingConfig/CreateRecording AND
- [S3] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet THEN <wsnt:TopicExpression> element contains tns1:RecordingConfig/CreateRecording OR tns1:RecordingConfig//. in expression AND

- Device response on the CreatePullPointSubscription request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] <soapenv:Body> element has child element <tev:CreatePullPointSubscriptionResponse>

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_RetrieveCreateRecordingNotifications

RETRIEVE DELETE RECORDING NOTIFICATION

Test Label: Dynamic Recording Configurations - Retrieve Notifications about deleted recordings

Test Case ID: DYNAMICRECORDINGCONFIGURATION-4

Profile G Requirement: Conditional

Feature Under Test: Dynamic Recording Configurations

Test Purpose: To verify that Client is able to retrieve notifications about deleted recordings from Device using the Pull Point event mechanism.

Pre-Requisite:

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

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

1. Client invokes CreatePullPointSubscription message to retrieve notification about deleted recording.

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

Test Result:

PASS -

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

- Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:
  - [S1] <soapenv:Body> element has child element <tev:CreatePullPointSubscription> AND
  - [S2] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete THEN <wsnt:TopicExpression> element is equal to tns1:RecordingConfig/DeleteRecording AND

- [S3] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet THEN <wsnt:TopicExpression> element contains tns1:RecordingConfig/DeleteRecording OR tns1:RecordingConfig//. in expression AND

- Device response on the CreatePullPointSubscription request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] <soapenv:Body> element has child element <tev:CreatePullPointSubscriptionResponse>

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_RetrieveDeleteRecordingNotifications

Recording Control – Dynamic Track Configurations Test Cases

Expected Scenarios Under Test:

1. Client subscribes to device messages using CreatePullPointSubscription operation to get notifications about created and deleted tracks.
2. Client creates new Track.
3. Client uses Pull Point event mechanism to retrieve notification events about created tracks from Device.
4. Client delete Track.
5. Client uses Pull Point event mechanism to retrieve notification events about deleted tracks from Device.
6. Client is considered as supporting Dynamic Tracks Configurations if the following conditions are met:
   - When Device supports DynamicTracks feature and Client supports Dynamic Tracks Configurations capability:
     - Client is able to create a track using the CreateTrack operation AND
     - Client is able to delete a track using the DeleteTrack operation AND
     - Client supports EventHandling_Pullpoint feature AND
     - Client is able to retrieve notifications about created tracks AND
     - Client is able to retrieve notifications about deleted tracks.
7. Client is considered as NOT supporting Dynamic Tracks Configurations if ANY of the following is TRUE:
   - When Device supports DynamicTracks feature and Client supports Dynamic Tracks Configurations capability:
     - No Valid Device Response to CreateTrack request (except soap fault: MaxTracks) OR
     - No Valid Device Response to DeleteTrack request (except soap fault: CannotDelete) OR

- Client does not support EventHandling_Pullpoint feature OR
- Client unable to retrieve notifications about created tracks OR
- Client unable to retrieve notifications about deleted tracks OR

RETRIEVE CREATE TRACK NOTIFICATION

Test Label: Dynamic Track Configurations - Retrieve Notifications about created tracks

Test Case ID: DYNAMICTRACKSCONFIGURATION-3

Profile G Requirement: Conditional

Feature Under Test: Dynamic Track Configurations

Test Purpose: To verify that Client is able to retrieve notifications about created tracks from Device using the Pull Point event mechanism.

Pre-Requisite:

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

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

1. Client invokes CreatePullPointSubscription message to retrieve notification about created tracks.
2. Device responses with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.

Test Result:

PASS -

- Client CreatePullPointSubscription request messages are valid according to XML Schemas listed in NamespacesAND
- Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:
  - [S1] <soapenv:Body> element has child element <tev:CreatePullPointSubscription> AND
  - [S2] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete THEN <wsnt:TopicExpression> element is equal to tns1:RecordingConfig/CreateTrack AND
  - [S3] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet THEN <wsnt:TopicExpression> element contains tns1:RecordingConfig/CreateTrack OR tns1:RecordingConfig//. in expression AND
- Device response on the CreatePullPointSubscription request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] <soapenv:Body> element has child element <tev:CreatePullPointSubscriptionResponse>

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_RetrieveCreateTrackNotifications

RETRIEVE DELETE TRACK NOTIFICATION

Test Label: Dynamic Track Configurations - Retrieve Notifications about deleted tracks

Test Case ID: DYNAMICTRACKSCONFIGURATION-4

Profile G Requirement: Conditional

Feature Under Test: Dynamic Track Configurations

Test Purpose: To verify that Client is able to retrieve notifications about deleted tracks from Device using the Pull Point event mechanism.

Pre-Requisite:

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

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

1. Client invokes CreatePullPointSubscription message to retrieve notification about deleted tracks.

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

Test Result:

PASS -

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

- Client CreatePullPointSubscription request in Test Procedure fulfills the following requirements:
  - [S1] <soapenv:Body> element has child element <tev:CreatePullPointSubscription> AND
  - [S2] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete THEN <wsnt:TopicExpression> element is equal to tns1:RecordingConfig/DeleteTrack AND
  - [S3] IF it contains <tcr:Filter> element with child element <wsnt:TopicExpression> with Dialect attribute equal to http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet THEN <wsnt:TopicExpression> element contains tns1:RecordingConfig/DeleteTrack OR tns1:RecordingConfig//. in expression AND

- Device response on the CreatePullPointSubscription request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] <soapenv:Body> element has child element <tev:CreatePullPointSubscriptionResponse>

FAIL -

- The Client failed PASS criteria.

Validated Feature List: DynamicRecordingConfigurations_RetrieveDeleteTrackNotifications

Video Sources List Test Cases

Validated Feature: get_video_sources

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to retrieve a video sources list.
2. Client is considered as supporting Video Sources List if the following conditions are met:
   - Client is able to retrieve a Video Sources List using **GetVideoSources** operation (Device IO Service OR Media Service).
3. Client is considered as NOT supporting Video Sources List if ANY of the following is TRUE:
   - No valid responses for **GetVideoSources** request (Device IO Service OR Media Service)

GET VIDEO SOURCES

Test Label: Video Sources List - Get Video Sources

Test Case ID: GETVIDEOSOURCES-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: None

Profile S Reference: None

Feature Under Test: Get Video Sources

Test Purpose: To verify that video sources for Device is received by Client using the **GetVideoSources** operation (Device IO Service OR Media Service).

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoSources** operation (Device IO Service OR Media Service) present.
- Device supports Imaging Service.

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

1. Client invokes **GetVideoSources** request message (Device IO Service OR Media Service) to retrieve list of video sources from the Device.
2. Device responses with code HTTP 200 OK and **GetVideoSourcesResponse** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: get_video_sources.get_video_sources

Remote User Handling Test Cases

Feature Level Requirement:

Validated Feature: remote_user_handling

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: None

Profile S Requirement: None

Expected Scenarios Under Test:

1. Client connects to Device to configure the credentials of a remote user.
2. Client is considered as supporting Remote User Handling if the following conditions are met:
   - Client is able to retrieve configured remote user using **GetRemoteUser** operation AND
   - Client is able to set the remote user using **SetRemoteUser** operation.
3. Client is considered as NOT supporting Remote User Handling if ANY of the following is TRUE:
   - No valid responses for **GetRemoteUser** request OR
   - No valid responses for **SetRemoteUser** request.

GET REMOTE USER

Test Label: Remote User Handling - Get Remote User

Test Case ID: REMOTEUSERHANDLING-1

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

Feature Under Test: Get Remote User

Test Purpose: To verify that Client is able to retrieve configured remote user of Device using the **GetRemoteUser** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetRemoteUser** operation present.
- Device supports Remote User Handling.

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

1. Client invokes **GetRemoteUser** request message to retrieve configured remote user from the Device.
2. Device responses with code HTTP 200 OK and **GetRemoteUser** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: remote_user_handling.get_remote_user

SET REMOTE USER

Test Label: Remote User Handling - Set Remote User

Test Case ID: REMOTEUSERHANDLING-2

Profile A Normative Reference: None

Profile C Normative Reference: None

Profile G Normative Reference: None

Profile Q Normative Reference: None

Profile S Normative Reference: None

Feature Under Test: Set Remote User

Test Purpose: To verify that Client is able to set the remote user using the **SetRemoteUser** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetRemoteUser** operation with **RemoteUser** element present.
- Device supports Remote User Handling.

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

1. Client invokes **SetRemoteUser** request message with **RemoteUser** element to configure remote user on the Device.
2. Device responses with code HTTP 200 OK and **SetRemoteUser** message.

Test Result:

PASS -

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

FAIL -

- The Client failed PASS criteria.

Validated Feature List: remote_user_handling.set_remote_user

A. Revision History

January 27, 2016 Version 1.0

- Remote User Handling Test Cases were moved from Core Client test spec since this functionality was removed from Profile Q

June 11, 2014 Version 1.0

- Initial version