



ONVIF Conformance Test

Performed by

Operator -

Organization -

Address -

Device Under Test

Product Name - onvif rtsp server

Brand - Happytimesoft

Model - IPCamera

Product Type - Encoder

Serial Number - 123456

Firmware Version - 2.4

Other -

ONVIF Device Test Tool version 22.06 rev. 7401

Test Date and Time - 2023/6/26 @ 12:11:12

ONVIF Test Summary

Tests Executed: 217

Tests Passed: 217

Tests Failed: 0

Features:

Discovery

Events

Media Service 2

IO

Imaging

Timeouts (ms):

Message Timeout: 5000

Reboot Timeout: 5000

Time between Tests: 1000

Time between Requests: 1000

Operation Delay: 10000

Real Timeouts (ms):

Maximum Timeout: 4937

Median Timeout: 16

Average Timeout: 65

Account: admin

TEST PASSED

Features Definition Log

DEFINE FEATURES

STEP 1 - GetCapabilities (no credentials supplied)

STEP PASSED

STEP 2 - GetServices (no credentials supplied)

STEP PASSED

STEP 3 - Check GetCapabilities and GetServices

STEP PASSED

STEP 4 - Get Service Capabilities (no credentials supplied)

STEP PASSED

STEP 5 - Analyze Device Service capabilities

STEP PASSED

STEP 6 - Define Network features

STEP PASSED

STEP 7 - Define Security capabilities

STEP PASSED

STEP 8 - Define System features

STEP PASSED

STEP 9 - Define Device IO features

STEP PASSED

STEP 10 - Define Misc features

STEP PASSED

STEP 11 - Get Relay Outputs

STEP PASSED

STEP 12 - Set Relay Output settings (IdleState=Monostable, Mode=closed)

STEP PASSED

STEP 13 - Set Relay Output settings (IdleState=Monostable, Mode=open)

STEP PASSED

STEP 14 - Set Relay Output settings (IdleState=Bistable, Mode=closed)

STEP PASSED

STEP 15 - Set Relay Output settings (IdleState=Bistable, Mode=open)

STEP PASSED

STEP 16 - Sending Unicast Probe request

STEP PASSED

STEP 17 - Define Discovery features

STEP PASSED

STEP 18 - Define Event service support

STEP PASSED

STEP 19 - Get Event service capabilities

STEP PASSED

STEP 20 - Define Media features

STEP PASSED

STEP 21 - Define Security Configuration support

STEP PASSED

STEP 22 - Get Network Protocols

STEP PASSED

STEP 23 - Define Media2 features

STEP PASSED

STEP 24 - Get Media2 Service Capabilities

STEP PASSED

STEP 25 - Get Media2 Video Encoder Configuration Options

STEP PASSED

STEP 26 - Get Video Source Configurations

STEP PASSED

STEP 27 - Get OSD Options

STEP PASSED

STEP 28 - Check IO service

STEP PASSED

STEP 29 - Get IO capabilities

STEP PASSED

STEP 30 - Define RelayOutputs features

STEP PASSED

STEP 31 - Get Relay Output Options

STEP PASSED

STEP 32 - Define Relay Output Options features

STEP PASSED

STEP 33 - Get Relay Outputs

STEP PASSED

Define Relay Output Node Features [token = RelayOutputToken_1]

STEP 34 - Define Relay Output Options of Relay Output Node [token = RelayOutputToken_1]

STEP PASSED

STEP 35 - Check that the DUT sent relay output options item with DelayTimes field when Monostable Mode is supported

STEP PASSED

STEP 36 - Set Relay Output Settings

STEP PASSED

STEP 37 - Check that the DUT sent relay output options item with DelayTimes field when Monostable Mode is supported

STEP PASSED

STEP 38 - Set Relay Output Settings

STEP PASSED

STEP 39 - Check that at least one Idle State is supported

STEP PASSED

STEP 40 - Set Relay Output Settings

STEP PASSED

STEP 41 - Set Relay Output Settings

STEP PASSED

STEP 42 - Check that at least one Idle State is supported

STEP PASSED

Overall state of Relay Output nodes features:

STEP 43 - Define DigitalInputs features

STEP PASSED

STEP 44 - Define DigitalInputOptions features

STEP PASSED

STEP 45 - Define PTZ service

STEP PASSED

STEP 46 - Define Imaging features

STEP PASSED

STEP 47 - Get Video Sources

STEP PASSED

STEP 48 - Get Imaging Options

STEP PASSED

STEP 49 - Define IrCutfilter feature

STEP PASSED

STEP 50 - Get Video Sources

STEP PASSED

STEP 51 - Get Move Options

STEP PASSED

STEP 52 - Define AnalyticsService features

STEP PASSED

STEP 53 - Define Recording Control service support
STEP PASSED

STEP 54 - Define Search service support
STEP PASSED

STEP 55 - Define Replay service support
STEP PASSED

STEP 56 - Define Receiver service support
STEP PASSED

STEP 57 - Define Access Rules support
STEP PASSED

STEP 58 - Define Credential support
STEP PASSED

STEP 59 - Define Schedule support
STEP PASSED

STEP 60 - Define AccessControl service support
STEP PASSED

STEP 61 - Define DoorControl service support
STEP PASSED

STEP 62 - Define Thermal service support

STEP PASSED

STEP 63 - Get Event Properties

STEP PASSED

STEP 64 - Define supported events

STEP PASSED

Define device scope(s)

STEP 65 - Get device scopes

STEP PASSED

STEP 66 - Check scopes

STEP PASSED

STEP 67 - Get device information

STEP PASSED

STEP 68 - Get Endpoint Address

STEP PASSED

STEP 69 - Check for Undefined features

STEP PASSED

PROCESS COMPLETED

Device Pre-Configuration Log

PRECONFIGURE DEVICE FOR CONFORMANCE

IPv6 is not supported, skipping IPv6 configuration procedure.

PROCESS COMPLETED

The following tests were FAILED:

Tests

MEDIA2_RTSS-1-1-1-v21.12 MEDIA2 STREAMING – H.264 (RTP-Unicast/UDP)
MEDIA2_RTSS-1-1-2-v21.06 MEDIA2 STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP)
MEDIA2_RTSS-1-1-3-v21.06 MEDIA2 STREAMING – H.264 (RTP/RTSP/TCP)
MEDIA2_RTSS-1-1-4-v21.12 MEDIA2 SET SYNCHRONIZATION POINT – H.264
MEDIA2_RTSS-1-1-8-v21.12 MEDIA2 STREAMING – H.265 (RTP-Unicast/UDP)
MEDIA2_RTSS-1-1-9-v21.06 MEDIA2 STREAMING – H.265 (RTP-Unicast/RTSP/HTTP/TCP)
MEDIA2_RTSS-1-1-10-v21.06 MEDIA2 STREAMING – H.265 (RTP/RTSP/TCP)
MEDIA2_RTSS-1-1-11-v21.12 MEDIA2 SET SYNCHRONIZATION POINT – H.265
MEDIA2_RTSS-1-1-23-v21.06 VIDEO ENCODER INSTANCES
MEDIA2_RTSS-1-1-24-v21.06 VIDEO ENCODER INSTANCES - H.264
MEDIA2_RTSS-1-1-25-v21.06 VIDEO ENCODER INSTANCES - H.265
MEDIA2_RTSS-1-2-1-v21.06 MEDIA2 STREAMING – H.264 (RTP-Multicast, IPv4)
MEDIA2_RTSS-1-2-3-v21.06 MEDIA2 STREAMING – H.265 (RTP-Multicast, IPv4)
MEDIA2_RTSS-4-1-1-v21.12 METADATA STREAMING (RTP-Unicast/UDP)
MEDIA2_RTSS-4-1-2-v20.12 METADATA STREAMING (RTP-Unicast/RTSP/HTTP/TCP)
MEDIA2_RTSS-4-1-3-v20.12 METADATA STREAMING (RTP/RTSP/TCP)
MEDIA2_RTSS-4-1-4-v21.12 METADATA STREAMING - SET SYNCHRONIZATION POINT
MEDIA2_RTSS-4-2-1-v20.12 METADATA STREAMING (RTP-Multicast/UDP)
IMAGING-1-1-1-v17.12 IMAGING COMMAND GETIMAGINGSETTINGS
IMAGING-1-1-3-v19.12 IMAGING COMMAND GETOPTIONS
IMAGING-1-1-8-v19.12 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID SETTINGS
IMAGING-1-1-10-v17.12 IMAGING COMMAND GETIMAGINGSETTINGS – INVALID
VIDEOSOURCETOKEN
IMAGING-1-1-11-v17.12 IMAGING COMMAND GETOPTIONS – INVALID VIDEOSOURCETOKEN
IMAGING-1-1-12-v17.12 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID
VIDEOSOURCETOKEN
IMAGING-1-1-14-v21.12 IMAGING COMMAND SETIMAGINGSETTINGS
IMAGING-1-1-15-v19.12 IMAGING COMMAND SETIMAGINGSETTINGS ADDITIONAL
FEATURES
IMAGING-1-1-16-v19.12 GET IMAGING SETTINGS AND GET OPTIONS CONSISTENCY
IMAGING-2-1-1-v17.12 IMAGING COMMAND GETMOVEOPTIONS
IMAGING-2-1-3-v17.12 IMAGING COMMAND ABSOLUTE MOVE
IMAGING-2-1-4-v17.12 IMAGING COMMAND ABSOLUTE MOVE – INVALID SETTINGS

IMAGING-2-1-5-v17.12 IMAGING COMMAND RELATIVE MOVE
IMAGING-2-1-6-v17.12 IMAGING COMMAND RELATIVE MOVE – INVALID SETTINGS
IMAGING-2-1-7-v17.12 IMAGING COMMAND CONTINUOUS MOVE
IMAGING-2-1-8-v17.12 IMAGING COMMAND CONTINUOUS MOVE – INVALID SETTINGS
IMAGING-2-1-10-v17.12 IMAGING COMMAND MOVE – UNSUPPORTED MOVE
IMAGING-2-1-11-v17.12 IMAGING COMMAND GETSTATUS
IMAGING-2-1-13-v17.12 IMAGING COMMAND STOP
IMAGING-2-1-15-v17.12 IMAGING COMMAND GETMOVEOPTIONS – INVALID
VIDEOSOURCETOKEN
IMAGING-2-1-16-v17.12 IMAGING COMMAND MOVE – INVALID VIDEOSOURCETOKEN
IMAGING-2-1-17-v17.12 IMAGING COMMAND GETSTATUS – INVALID VIDEOSOURCETOKEN
IMAGING-2-1-18-v17.12 IMAGING COMMAND STOP – INVALID VIDEOSOURCETOKEN
IMAGING-3-1-1-v14.12 IMAGING SERVICE CAPABILITIES
IMAGING-3-1-2-v14.12 GET SERVICES AND GET IMAGING SERVICE CAPABILITIES
CONSISTENCY
IMAGING-4-1-1-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO BLURRY
IMAGING-4-1-2-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO DARK
IMAGING-4-1-3-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO BRIGHT
IMAGING-4-1-4-v18.06 REALTIME PULLPOINT SUBSCRIPTION – GLOBAL SCENE CHANGE
IMAGING-4-1-5-v18.06 REALTIME PULLPOINT SUBSCRIPTION – MOTION ALARM
DEVICEIO-1-1-1-v16.07 IO GETRELAYOUTPUTS
DEVICEIO-1-1-2-v17.12 IO GETRELAYOUTPUTS – VERIFY QUANTITY
DEVICEIO-1-1-3-v16.07 IO GETRELAYOUTPUTOPTIONS
DEVICEIO-1-1-4-v18.06SR1 IO SETRELAYOUTPUTSETTINGS
DEVICEIO-1-1-5-v16.07 IO SETRELAYOUTPUTSETTINGS – INVALID TOKEN
DEVICEIO-1-2-1-v18.06SR1 IO SETRELAYOUTPUTSTATE – BISTABLE MODE (OPENED IDLE
STATE)
DEVICEIO-1-2-2-v18.06SR1 IO SETRELAYOUTPUTSTATE – BISTABLE MODE (CLOSED IDLE
STATE)
DEVICEIO-1-2-3-v19.12 IO SETRELAYOUTPUTSTATE – MONOSTABLE MODE (OPENED IDLE
STATE)
DEVICEIO-1-2-4-v19.12 IO SETRELAYOUTPUTSTATE – MONOSTABLE MODE (CLOSED IDLE
STATE)
DEVICEIO-2-1-1-v18.06 REALTIME PULLPOINT SUBSCRIPTION – DIGITAL INPUT EVENT
DEVICEIO-2-1-2-v17.12 DEVICE IO SERVICE TRIGGER EVENT CHECK

DEVICEIO-3-1-1-v17.01 GETDIGITALINPUTS
DEVICEIO-3-1-2-v17.01 GETDIGITALINPUTS – VERIFY QUANTITY
DEVICEIO-3-1-3-v17.12 I/O GET DIGITAL INPUT CONFIGURATION OPTIONS
DEVICEIO-3-1-4-v17.12 I/O DIGITAL INPUT CONFIGURATION
DEVICEIO-7-1-1-v17.12 IO GET VIDEO SOURCES
MEDIA2-1-1-1-v17.06 READY TO USE MEDIA PROFILE FOR VIDEO STREAMING
MEDIA2-1-1-2-v20.06 CREATE MEDIA PROFILE WITH PRE-DEFINED CONFIGURATION
MEDIA2-1-1-3-v20.12 DYNAMIC MEDIA PROFILE CONFIGURATION
MEDIA2-1-1-4-v19.12 GET PROFILES
MEDIA2-1-1-5-v20.12 CREATE MEDIA PROFILE WITH CONFIGURATIONS
MEDIA2-1-1-6-v20.06 REMOVE ALL CONFIGURATIONS FROM MEDIA PROFILE
MEDIA2-1-1-7-v20.06 FIXED MEDIA PROFILE CONFIGURATION
MEDIA2-2-2-1-v20.06 GET VIDEO SOURCE CONFIGURATION OPTIONS
MEDIA2-2-2-2-v17.01 GET VIDEO SOURCE CONFIGURATIONS
MEDIA2-2-2-3-v17.01 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE
CONFIGURATION OPTIONS CONSISTENCY
MEDIA2-2-2-4-v17.01 PROFILES AND VIDEO SOURCE CONFIGURATIONS CONSISTENCY
MEDIA2-2-2-5-v21.12 MODIFY ALL SUPPORTED VIDEO SOURCE CONFIGURATIONS
MEDIA2-2-2-6-v17.01 GET VIDEO SOURCE CONFIGURATIONS – INVALID TOKEN
MEDIA2-2-2-7-v17.12 PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS
CONSISTENCY
MEDIA2-2-3-1-v20.12 VIDEO ENCODER CONFIGURATION
MEDIA2-2-3-2-v20.12 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER
CONFIGURATION OPTIONS CONSISTENCY VALIDATION
MEDIA2-2-3-3-v20.12 PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS
CONSISTENCY VALIDATION
MEDIA2-2-3-4-v20.12 SET ALL SUPPORTED VIDEO ENCODER CONFIGURATIONS
MEDIA2-2-3-5-v20.12 VIDEO ENCODER CONFIGURATION OPTIONS VALIDATION
MEDIA2-2-4-1-v17.01 GET VIDEO SOURCE MODES
MEDIA2-2-4-2-v21.06 SET VIDEO SOURCE MODES
MEDIA2-5-1-1-v20.12 SNAPSHOT URI
MEDIA2-5-1-2-v20.12 VIDEO ENCODER INSTANCES PER VIDEO SOURCE
MEDIA2-6-1-1-v18.06 CREATE OSD CONFIGURATION FOR TEXT OVERLAY
MEDIA2-6-1-2-v20.06 CREATE OSD CONFIGURATION FOR IMAGE OVERLAY
MEDIA2-6-1-3-v20.06 SET OSD CONFIGURATION IMAGE OVERLAY

MEDIA2-6-1-4-v18.06 SET OSD CONFIGURATION TEXT OVERLAY
MEDIA2-6-1-5-v17.12 GET OSDS
MEDIA2-6-1-6-v18.06 GET OSD OPTIONS
MEDIA2-6-1-7-v18.06 OSD CONFIGURATIONS AND OSD OPTIONS CONSISTENCY
MEDIA2-7-1-1-v18.12 MEDIA2 SERVICE CAPABILITIES
MEDIA2-7-1-2-v17.06 GET SERVICES AND GET MEDIA2 SERVICE CAPABILITIES CONSISTENCY
MEDIA2-8-1-1-v20.12 MODIFY ALL SUPPORTED METADATA CONFIGURATIONS
MEDIA2-8-1-2-v19.12 GET METADATA CONFIGURATIONS
MEDIA2-8-1-3-v19.12 PROFILES AND METADATA CONFIGURATIONS CONSISTENCY
MEDIA2-8-1-4-v19.12 GET METADATA CONFIGURATIONS – INVALID TOKEN
SECURITY-1-1-1-v14.12 USER TOKEN PROFILE
SECURITY-1-1-2-v14.12 DIGEST AUTHENTICATION
IPCONFIG-1-1-3-v21.06 IPV4 DHCP
IPCONFIG-1-1-5-v20.12 IPV4 LINK LOCAL ADDRESS
DISCOVERY-1-1-2-v21.06 HELLO MESSAGE VALIDATION
DISCOVERY-1-1-3-v21.06 SEARCH BASED ON DEVICE SCOPE TYPES
DISCOVERY-1-1-4-v21.06 SEARCH WITH OMITTED DEVICE AND SCOPE TYPES
DISCOVERY-1-1-5-v21.06 RESPONSE TO INVALID SEARCH REQUEST
DISCOVERY-1-1-6-v21.06 SEARCH USING UNICAST PROBE MESSAGE
DISCOVERY-1-1-8-v14.12 BYE MESSAGE
DISCOVERY-1-1-9-v21.06 DISCOVERY MODE CONFIGURATION
DISCOVERY-1-1-11-v21.06 DEVICE SCOPES CONFIGURATION
DISCOVERY-2-1-1-v21.06 DISCOVERY - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)
DISCOVERY-2-1-2-v21.06 DISCOVERY - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)
DISCOVERY-2-1-3-v21.06 DISCOVERY - NAMESPACES (NOT STANDARD PREFIXES)
DISCOVERY-2-1-4-v21.06 DISCOVERY - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)
DISCOVERY-2-1-5-v21.06 DISCOVERY - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)
DEVICE-1-1-2-v14.12 ALL CAPABILITIES
DEVICE-1-1-3-v14.12 DEVICE CAPABILITIES
DEVICE-1-1-4-v14.12 MEDIA CAPABILITIES
DEVICE-1-1-5-v14.12 EVENT CAPABILITIES

DEVICE-1-1-6-v14.12 PTZ CAPABILITIES
DEVICE-1-1-9-v14.12 SOAP FAULT MESSAGE
DEVICE-1-1-10-v14.12 IMAGING CAPABILITIES
DEVICE-1-1-11-v14.12 ANALYTICS CAPABILITIES
DEVICE-1-1-13-v14.12 GET SERVICES – DEVICE SERVICE
DEVICE-1-1-16-v14.12 GET SERVICES – EVENT SERVICE
DEVICE-1-1-17-v14.12 GET SERVICES – IMAGING SERVICE
DEVICE-1-1-18-v21.06 DEVICE SERVICE CAPABILITIES
DEVICE-1-1-19-v21.06 GET SERVICES AND GET DEVICE SERVICE CAPABILITIES
CONSISTENCY
DEVICE-1-1-30-v17.06 GET SERVICES AND GET CAPABILITIES CONSISTENCY
DEVICE-1-1-31-v18.12 GET SERVICES - XADDR
DEVICE-2-1-1-v20.12 NETWORK COMMAND HOSTNAME CONFIGURATION
DEVICE-2-1-3-v20.12 NETWORK COMMAND SETHOSTNAME TEST ERROR CASE
DEVICE-2-1-4-v20.12 GET DNS CONFIGURATION
DEVICE-2-1-5-v14.12 SET DNS CONFIGURATION - SEARCHDOMAIN
DEVICE-2-1-6-v21.06 SET DNS CONFIGURATION - DNSMANUAL IPV4
DEVICE-2-1-8-v21.06 SET DNS CONFIGURATION - FROMDHCP
DEVICE-2-1-11-v20.12 GET NTP CONFIGURATION
DEVICE-2-1-12-v21.06 SET NTP CONFIGURATION - NTPMANUAL IPV4
DEVICE-2-1-14-v21.06 SET NTP CONFIGURATION - FROMDHCP
DEVICE-2-1-17-v20.12 GET NETWORK INTERFACE CONFIGURATION
DEVICE-2-1-18-v21.06 SET NETWORK INTERFACE CONFIGURATION - IPV4
DEVICE-2-1-25-v20.12 GET NETWORK DEFAULT GATEWAY CONFIGURATION
DEVICE-2-1-30-v21.06 SET NETWORK DEFAULT GATEWAY CONFIGURATION - IPV4
DEVICE-2-1-32-v20.12 NETWORK COMMAND SETHOSTNAME TEST
DEVICE-2-1-33-v20.12 GET NETWORK PROTOCOLS CONFIGURATION
DEVICE-2-1-34-v20.12 SET NETWORK PROTOCOLS CONFIGURATION
DEVICE-2-1-35-v20.12 SET NETWORK PROTOCOLS CONFIGURATION - UNSUPPORTED
PROTOCOLS
DEVICE-2-1-36-v20.12 GET DYNAMIC DNS CONFIGURATION
DEVICE-3-1-1-v14.12 SYSTEM COMMAND GETSYSTEMDATEANDTIME
DEVICE-3-1-4-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST FOR INVALID
TIMEZONE
DEVICE-3-1-5-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST FOR INVALID

DATE

DEVICE-3-1-7-v21.06 SYSTEM COMMAND FACTORY DEFAULT SOFT

DEVICE-3-1-8-v21.06 SYSTEM COMMAND REBOOT

DEVICE-3-1-9-v14.12 SYSTEM COMMAND DEVICE INFORMATION

DEVICE-3-1-10-v14.12 SYSTEM COMMAND GETSYSTEMLOG

DEVICE-3-1-11-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME

DEVICE-3-1-12-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME USING NTP

DEVICE-3-1-13-v20.06 GET SYSTEM URIS

DEVICE-3-1-14-v21.06 START SYSTEM RESTORE

DEVICE-3-1-15-v21.06 START SYSTEM RESTORE – INVALID BACKUP FILE

DEVICE-4-1-1-v20.12 SECURITY COMMAND GETUSERS

DEVICE-4-1-3-v20.12 SECURITY COMMAND CREATEUSERS ERROR CASE

DEVICE-4-1-4-v20.12 SECURITY COMMAND DELETEUSERS

DEVICE-4-1-5-v20.12 SECURITY COMMAND DELETEUSERS ERROR CASE

DEVICE-4-1-7-v20.12 SECURITY COMMAND SETUSER

DEVICE-4-1-8-v20.12 SECURITY COMMAND USER MANAGEMENT ERROR CASE

DEVICE-4-1-9-v20.12 SECURITY COMMAND CREATEUSERS

DEVICE-4-1-10-v14.12 GET REMOTE USER

DEVICE-4-1-11-v14.12 SET REMOTE USER

DEVICE-5-1-1-v16.07 IO COMMAND GETRELAYOUTPUTS

DEVICE-5-1-2-v16.07 RELAY OUTPUTS COUNT IN GETRELAYOUTPUTS AND
GETCAPABILITIES

DEVICE-5-1-3-v16.07 IO COMMAND SETRELAYOUTPUTSETTINGS

DEVICE-5-1-5-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – BISTABLE MODE (OPENED
IDLE STATE)

DEVICE-5-1-6-v14.12 IO COMMAND SETRELAYOUTPUTSTATE – BISTABLE MODE (CLOSED
IDLE STATE)

DEVICE-5-1-7-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(OPENED IDLE STATE)

DEVICE-5-1-8-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(CLOSED IDLE STATE)

DEVICE-5-1-9-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(INACTIVE BEFORE DELAYTIME EXPIRED)

DEVICE-5-1-11-v16.07 IO COMMAND SETRELAYOUTPUTSETTINGS – INVALID TOKEN

DEVICE-5-1-12-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – INVALID TOKEN

DEVICE-6-1-1-v21.06 DEVICE MANAGEMENT - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)

DEVICE-6-1-2-v21.06 DEVICE MANAGEMENT - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)

DEVICE-6-1-3-v21.06 DEVICE MANAGEMENT - NAMESPACES (NOT STANDARD PREFIXES)

DEVICE-6-1-4-v21.06 DEVICE MANAGEMENT - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)

DEVICE-6-1-5-v21.06 DEVICE MANAGEMENT - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)

DEVICE-8-1-1-v17.01 AUXILIARY COMMANDS

EVENT-1-1-2-v19.06 GET EVENT PROPERTIES

EVENT-2-1-9-v14.12 BASIC NOTIFICATION INTERFACE - SUBSCRIBE

EVENT-2-1-12-v14.12 BASIC NOTIFICATION INTERFACE - RENEW

EVENT-2-1-17-v14.12 BASIC NOTIFICATION INTERFACE - NOTIFY

EVENT-2-1-18-v14.12 BASIC NOTIFICATION INTERFACE - NOTIFY FILTER

EVENT-2-1-24-v17.06 BASIC NOTIFICATION INTERFACE - SET SYNCHRONIZATION POINT

EVENT-2-1-25-v17.06 BASIC NOTIFICATION INTERFACE – CONJUNCTION IN NOTIFY FILTER (OR OPERATION)

EVENT-2-1-26-v17.06 BASIC NOTIFICATION INTERFACE – TOPIC SUB-TREE IN PULLMESSAGES FILTER

EVENT-2-1-27-v17.06 BASIC NOTIFICATION INTERFACE – CONJUNCTION IN NOTIFY FILTER (TOPIC SUB-TREE AND OR OPERATION)

EVENT-2-1-28-v17.12 BASIC NOTIFICATION INTERFACE - UNSUBSCRIBE

EVENT-2-1-29-v18.06 BASIC NOTIFICATION INTERFACE - MESSAGE CONTENT FILTER

EVENT-3-1-9-v14.12 REALTIME PULLPOINT SUBSCRIPTION - CREATE PULL POINT SUBSCRIPTION

EVENT-3-1-15-v14.12 REALTIME PULLPOINT SUBSCRIPTION - PULLMESSAGES

EVENT-3-1-16-v21.06 REALTIME PULLPOINT SUBSCRIPTION - PULLMESSAGES FILTER

EVENT-3-1-24-v14.12 REALTIME PULLPOINT SUBSCRIPTION – PULLMESSAGES AS KEEP-ALIVE

EVENT-3-1-25-v17.06 REALTIME PULLPOINT SUBSCRIPTION – SET SYNCHRONIZATION POINT

EVENT-3-1-32-v17.06 REALTIME PULLPOINT SUBSCRIPTION – PULLMESSAGES TIMEOUT

EVENT-3-1-33-v21.06 REALTIME PULLPOINT SUBSCRIPTION – CONJUNCTION IN PULLMESSAGES FILTER (OR OPERATION)

EVENT-3-1-34-v21.06 REALTIME PULLPOINT SUBSCRIPTION – TOPIC SUB-TREE IN

PULLMESSAGES FILTER

EVENT-3-1-35-v21.06 REALTIME PULLPOINT SUBSCRIPTION – CONJUNCTION IN NOTIFY FILTER (TOPIC SUB-TREE AND OR OPERATION)

EVENT-3-1-36-v17.12 REALTIME PULLPOINT SUBSCRIPTION - UNSUBSCRIBE

EVENT-3-1-37-v17.12 REALTIME PULLPOINT SUBSCRIPTION – MAXIMUM SUPPORTED NUMBER OF NOTIFICATION PULL POINTS

EVENT-3-1-38-v18.06 REALTIME PULLPOINT SUBSCRIPTION - MESSAGE CONTENT FILTER

EVENT-4-1-6-v16.07 EVENT - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)

EVENT-4-1-7-v16.07 EVENT - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)

EVENT-4-1-8-v16.07 EVENT - NAMESPACES (NOT STANDARD PREFIXES)

EVENT-4-1-9-v16.07 EVENT - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)

EVENT-4-1-10-v16.07 EVENT - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)

EVENT-5-1-1-v20.06 EVENT SERVICE CAPABILITIES

EVENT-5-1-2-v20.06 GET SERVICES AND EVENT SERVICE CAPABILITIES CONSISTENCY

ONVIF TEST

Real Time Streaming using Media2

MEDIA2_RTSS-1-1-1-v21.12 MEDIA2 STREAMING – H.264 (RTP-Unicast/UDP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1,
ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtspsUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Checking filters

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 21 - Teardown

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-2-v21.06 MEDIA2 STREAMING – H.264 (RTP-Unicast/RTSP/HTTP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtpOverHttp, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the same port with the web service

STEP PASSED

STEP 13 - Check if the stream uri has the same scheme with the web service

STEP PASSED

STEP 14 - Checking filters

STEP PASSED

STEP 15 - Describe

STEP PASSED

STEP 16 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 17 - Create Media Session

STEP PASSED

STEP 18 - Setup

STEP PASSED

STEP 19 - Create Sinks

STEP PASSED

STEP 20 - Play

STEP PASSED

STEP 21 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 22 - Teardown

STEP PASSED

STEP 23 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 24 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 25 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-3-v21.06 MEDIA2 STREAMING – H.264 (RTP/RTSP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RTSP, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Checking filters

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 21 - Teardown

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-4-v21.12 MEDIA2 SET SYNCHRONIZATION POINT – H.264

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Describe

STEP PASSED

STEP 14 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 15 - Create Media Session

STEP PASSED

STEP 16 - Setup

STEP PASSED

STEP 17 - Create Sinks

STEP PASSED

STEP 18 - Play

STEP PASSED

STEP 19 - Waiting for 10 seconds

STEP PASSED

STEP 20 - Teardown

STEP PASSED

STEP 21 - Checking media frames count

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-8-v21.12 MEDIA2 STREAMING – H.265 (RTP-Unicast/UDP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Checking filters

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 21 - Teardown

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-9-v21.06 MEDIA2 STREAMING – H.265 (RTP-Unicast/RTSP/HTTP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1,
ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtspsOverHttp, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the same port with the web service

STEP PASSED

STEP 13 - Check if the stream uri has the same scheme with the web service

STEP PASSED

STEP 14 - Checking filters

STEP PASSED

STEP 15 - Describe

STEP PASSED

STEP 16 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 17 - Create Media Session

STEP PASSED

STEP 18 - Setup

STEP PASSED

STEP 19 - Create Sinks

STEP PASSED

STEP 20 - Play

STEP PASSED

STEP 21 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 22 - Teardown

STEP PASSED

STEP 23 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 24 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 25 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-10-v21.06 MEDIA2 STREAMING – H.265 (RTP/RTSP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1,
ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RTSP, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Checking filters

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 21 - Teardown

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-11-v21.12 MEDIA2 SET SYNCHRONIZATION POINT – H.265

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 9 - Get Stream Uri (Media2) [Protocol = RtspsUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 10 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 11 - Check if the stream uri has correct IP type

STEP PASSED

STEP 12 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 13 - Describe

STEP PASSED

STEP 14 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 15 - Create Media Session

STEP PASSED

STEP 16 - Setup

STEP PASSED

STEP 17 - Create Sinks

STEP PASSED

STEP 18 - Play

STEP PASSED

STEP 19 - Waiting for 10 seconds

STEP PASSED

STEP 20 - Teardown

STEP PASSED

STEP 21 - Checking media frames count

STEP PASSED

STEP 22 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-23-v21.06 VIDEO ENCODER INSTANCES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Service Capabilities(Media2)

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Get Video Encoder Instances (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1]

STEP PASSED

STEP 11 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 12 - Check the number of Media Profiles to be created less than difference between MaximumNumberOfProfiles in ProfileCapabilities and number of fixed Media Profiles

STEP PASSED

STEP 13 - Check the number of Media Profiles to be created less than number of Video Encoder Configurations

STEP PASSED

STEP 14 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { All }]

STEP PASSED

STEP 15 - Remove Configuration (Media2) [ProfileToken = ProfileToken_2, Configuration = { All }]

STEP PASSED

STEP 16 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 17 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_3]

STEP PASSED

STEP 18 - Check that item with videoSourceConfig1.@token is presented in VideoSourceConfigurations

STEP PASSED

STEP 19 - Add Configuration (Media2) [ProfileToken = ProfileToken_3, no Name, Configuration = { VideoSource
(VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 20 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_3]

STEP PASSED

STEP 21 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 22 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured
Media Profiles list

STEP PASSED

STEP 23 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1,
ProfileToken = ProfileToken_3]

STEP PASSED

STEP 24 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 25 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 26 - Add Configuration (Media2) [ProfileToken = ProfileToken_3, no Name, Configuration = { VideoEncoder
(VideoEncoderConfigurationToken_1) }]

STEP PASSED

STEP 27 - Get Profiles (Media2) [Token = ProfileToken_3, Type = { All }]

STEP PASSED

STEP 28 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 29 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_4]

STEP PASSED

STEP 30 - Check that item with videoSourceConfig1.@token is presented in VideoSourceConfigurations

STEP PASSED

STEP 31 - Add Configuration (Media2) [ProfileToken = ProfileToken_4, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 32 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_4]

STEP PASSED

STEP 33 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 34 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured Media Profiles list

STEP PASSED

STEP 35 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, ProfileToken = ProfileToken_4]

STEP PASSED

STEP 36 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 37 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 38 - Add Configuration (Media2) [ProfileToken = ProfileToken_4, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

STEP 39 - Get Profiles (Media2) [Token = ProfileToken_4, Type = { All }]

STEP PASSED

STEP 40 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_3]

STEP PASSED

STEP 41 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 42 - Check if the stream uri has correct IP type

STEP PASSED

STEP 43 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 44 - [Profile: ProfileToken_3] Describe

STEP PASSED

STEP 45 - [Profile: ProfileToken_3] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 46 - [Profile: ProfileToken_3] Create Media Session

STEP PASSED

STEP 47 - [Profile: ProfileToken_3] Setup

STEP PASSED

STEP 48 - [Profile: ProfileToken_3] Create Sinks

STEP PASSED

STEP 49 - [Profile: ProfileToken_3] Play

STEP PASSED

STEP 50 - Get Stream Uri (Media2) [Protocol = RtspUnicast, ProfileToken = ProfileToken_4]

STEP PASSED

STEP 51 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 52 - Check if the stream uri has correct IP type

STEP PASSED

STEP 53 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 54 - [Profile: ProfileToken_4] Describe

STEP PASSED

STEP 55 - [Profile: ProfileToken_4] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 56 - [Profile: ProfileToken_4] Create Media Session

STEP PASSED

STEP 57 - [Profile: ProfileToken_4] Setup

STEP PASSED

STEP 58 - [Profile: ProfileToken_4] Create Sinks

STEP PASSED

STEP 59 - [Profile: ProfileToken_4] Play

STEP PASSED

STEP 60 - Closing streams

STEP 61 - [Profile: ProfileToken_3] Teardown

STEP PASSED

STEP 62 - [Profile: ProfileToken_4] Teardown

STEP PASSED

STEP PASSED

STEP 63 - Check for test results

STEP PASSED

STEP 64 - Delete Profile (Media2) [Token = ProfileToken_3]

STEP PASSED

STEP 65 - Delete Profile (Media2) [Token = ProfileToken_4]

STEP PASSED

STEP 66 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 67 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 68 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 69 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 70 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 71 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 72 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

Restore profile 'ProfileToken_2' used for test

STEP 73 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 74 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 75 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 76 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 77 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-24-v21.06 VIDEO ENCODER INSTANCES - H.264

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Service Capabilities(Media2)

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Get Video Encoder Instances (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1]

STEP PASSED

STEP 11 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 12 - Check the number of Media Profiles to be created less than difference between MaximumNumberOfProfiles in ProfileCapabilities and number of fixed Media Profiles

STEP PASSED

STEP 13 - Check the number of Media Profiles to be created less than number of Video Encoder Configurations

STEP PASSED

STEP 14 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { All }]

STEP PASSED

STEP 15 - Remove Configuration (Media2) [ProfileToken = ProfileToken_2, Configuration = { All }]

STEP PASSED

Number of media profiles to be configured for VideoSource [token = VideoSourceConfigurationToken_1]: 2.

STEP 16 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 17 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_5]

STEP PASSED

STEP 18 - Check that item with token VideoSourceConfigurationToken_1 is presented in VideoSourceConfigurations

STEP PASSED

STEP 19 - Add Configuration (Media2) [ProfileToken = ProfileToken_5, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 20 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_5]

STEP PASSED

STEP 21 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 22 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured Media Profiles list

STEP PASSED

STEP 23 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_5]

STEP PASSED

STEP 24 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 25 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 26 - Add Configuration (Media2) [ProfileToken = ProfileToken_5, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

STEP 27 - Get Profiles (Media2) [Token = ProfileToken_5, Type = { All }]

STEP PASSED

STEP 28 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 29 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_6]

STEP PASSED

STEP 30 - Check that item with token VideoSourceConfigurationToken_1 is presented in VideoSourceConfigurations

STEP PASSED

STEP 31 - Add Configuration (Media2) [ProfileToken = ProfileToken_6, no Name, Configuration = { VideoSource
(VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 32 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_6]

STEP PASSED

STEP 33 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 34 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured
Media Profiles list

STEP PASSED

STEP 35 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2,
ProfileToken = ProfileToken_6]

STEP PASSED

STEP 36 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 37 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 38 - Add Configuration (Media2) [ProfileToken = ProfileToken_6, no Name, Configuration = { VideoEncoder
(VideoEncoderConfigurationToken_2) }]

STEP PASSED

STEP 39 - Get Profiles (Media2) [Token = ProfileToken_6, Type = { All }]

STEP PASSED

Media profiles for VideoSource [token = VideoSourceConfigurationToken_1] was configured.

Profiles configured for VideoSource [token = VideoSourceConfigurationToken_1]:

Guaranteed encoder instances:

Total = 2

Configured media profiles:

Profile token = ProfileToken_5

VEC token = VideoEncoderConfigurationToken_1

Encoding = H264

Profile token = ProfileToken_6

VEC token = VideoEncoderConfigurationToken_2

Encoding = H264

STEP 40 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_5]

STEP PASSED

STEP 41 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 42 - Check if the stream uri has correct IP type

STEP PASSED

STEP 43 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 44 - [Profile: ProfileToken_5] Describe

STEP PASSED

STEP 45 - [Profile: ProfileToken_5] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 46 - [Profile: ProfileToken_5] Create Media Session

STEP PASSED

STEP 47 - [Profile: ProfileToken_5] Setup

STEP PASSED

STEP 48 - [Profile: ProfileToken_5] Create Sinks

STEP PASSED

STEP 49 - [Profile: ProfileToken_5] Play

STEP PASSED

STEP 50 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_6]

STEP PASSED

STEP 51 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 52 - Check if the stream uri has correct IP type

STEP PASSED

STEP 53 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 54 - [Profile: ProfileToken_6] Describe

STEP PASSED

STEP 55 - [Profile: ProfileToken_6] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 56 - [Profile: ProfileToken_6] Create Media Session

STEP PASSED

STEP 57 - [Profile: ProfileToken_6] Setup

STEP PASSED

STEP 58 - [Profile: ProfileToken_6] Create Sinks

STEP PASSED

STEP 59 - [Profile: ProfileToken_6] Play

STEP PASSED

STEP 60 - Closing streams

STEP 61 - [Profile: ProfileToken_5] Teardown

STEP PASSED

STEP 62 - [Profile: ProfileToken_6] Teardown

STEP PASSED

STEP PASSED

STEP 63 - Check for test results

STEP PASSED

STEP 64 - Delete Profile (Media2) [Token = ProfileToken_5]

STEP PASSED

STEP 65 - Delete Profile (Media2) [Token = ProfileToken_6]

STEP PASSED

STEP 66 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 67 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 68 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 69 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 70 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 71 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 72 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

Restore profile 'ProfileToken_2' used for test

STEP 73 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 74 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 75 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 76 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 77 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-1-25-v21.06 VIDEO ENCODER INSTANCES - H.265

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Service Capabilities(Media2)

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Get Video Encoder Instances (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1]

STEP PASSED

STEP 11 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 12 - Check the number of Media Profiles to be created less than difference between MaximumNumberOfProfiles in ProfileCapabilities and number of fixed Media Profiles

STEP PASSED

STEP 13 - Check the number of Media Profiles to be created less than number of Video Encoder Configurations

STEP PASSED

STEP 14 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { All }]

STEP PASSED

STEP 15 - Remove Configuration (Media2) [ProfileToken = ProfileToken_2, Configuration = { All }]

STEP PASSED

Number of media profiles to be configured for VideoSource [token = VideoSourceConfigurationToken_1]: 2.

STEP 16 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 17 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_7]

STEP PASSED

STEP 18 - Check that item with token VideoSourceConfigurationToken_1 is presented in VideoSourceConfigurations

STEP PASSED

STEP 19 - Add Configuration (Media2) [ProfileToken = ProfileToken_7, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 20 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_7]

STEP PASSED

STEP 21 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 22 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured Media Profiles list

STEP PASSED

STEP 23 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_7]

STEP PASSED

STEP 24 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 25 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 26 - Add Configuration (Media2) [ProfileToken = ProfileToken_7, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

STEP 27 - Get Profiles (Media2) [Token = ProfileToken_7, Type = { All }]

STEP PASSED

STEP 28 - Create Profile (Media2) [Name = testMedia, no Configuration]

STEP PASSED

STEP 29 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_8]

STEP PASSED

STEP 30 - Check that item with token VideoSourceConfigurationToken_1 is presented in VideoSourceConfigurations

STEP PASSED

STEP 31 - Add Configuration (Media2) [ProfileToken = ProfileToken_8, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 32 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_8]

STEP PASSED

STEP 33 - Check list of VideoEncoderConfiguration items is not empty

STEP PASSED

STEP 34 - Check list of VideoEncoderConfiguration items contains only items that were used in Media Profiles from Configured Media Profiles list

STEP PASSED

STEP 35 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, ProfileToken = ProfileToken_8]

STEP PASSED

STEP 36 - Check the appropriate VideoEncoderConfiguration found

STEP PASSED

STEP 37 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 38 - Add Configuration (Media2) [ProfileToken = ProfileToken_8, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

STEP 39 - Get Profiles (Media2) [Token = ProfileToken_8, Type = { All }]

STEP PASSED

Media profiles for VideoSource [token = VideoSourceConfigurationToken_1] was configured.

Profiles configured for VideoSource [token = VideoSourceConfigurationToken_1]:

Guaranteed encoder instances:

Total = 2

Configured media profiles:

Profile token = ProfileToken_7

VEC token = VideoEncoderConfigurationToken_1

Encoding = H265

Profile token = ProfileToken_8

VEC token = VideoEncoderConfigurationToken_2

Encoding = H265

STEP 40 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_7]

STEP PASSED

STEP 41 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 42 - Check if the stream uri has correct IP type

STEP PASSED

STEP 43 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 44 - [Profile: ProfileToken_7] Describe

STEP PASSED

STEP 45 - [Profile: ProfileToken_7] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 46 - [Profile: ProfileToken_7] Create Media Session

STEP PASSED

STEP 47 - [Profile: ProfileToken_7] Setup

STEP PASSED

STEP 48 - [Profile: ProfileToken_7] Create Sinks

STEP PASSED

STEP 49 - [Profile: ProfileToken_7] Play

STEP PASSED

STEP 50 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_8]

STEP PASSED

STEP 51 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 52 - Check if the stream uri has correct IP type

STEP PASSED

STEP 53 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 54 - [Profile: ProfileToken_8] Describe

STEP PASSED

STEP 55 - [Profile: ProfileToken_8] Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 56 - [Profile: ProfileToken_8] Create Media Session

STEP PASSED

STEP 57 - [Profile: ProfileToken_8] Setup

STEP PASSED

STEP 58 - [Profile: ProfileToken_8] Create Sinks

STEP PASSED

STEP 59 - [Profile: ProfileToken_8] Play

STEP PASSED

STEP 60 - Closing streams

STEP 61 - [Profile: ProfileToken_7] Teardown

STEP PASSED

STEP 62 - [Profile: ProfileToken_8] Teardown

STEP PASSED

STEP PASSED

STEP 63 - Check for test results

STEP PASSED

STEP 64 - Delete Profile (Media2) [Token = ProfileToken_7]

STEP PASSED

STEP 65 - Delete Profile (Media2) [Token = ProfileToken_8]

STEP PASSED

STEP 66 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 67 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 68 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 69 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 70 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 71 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 72 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

Restore profile 'ProfileToken_2' used for test

STEP 73 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 74 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 75 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 76 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 77 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-2-1-v21.06 MEDIA2 STREAMING – H.264 (RTP-Multicast, IPv4)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 8 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { AudioEncoder, Metadata }]

STEP PASSED

STEP 9 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 10 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { Metadata }]

STEP PASSED

STEP 11 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 12 - Get Stream Uri (Media2) [Protocol = RtpMulticast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 13 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 14 - Check if the stream uri has correct IP type

STEP PASSED

STEP 15 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 16 - Checking filters

STEP PASSED

STEP 17 - Describe

STEP PASSED

STEP 18 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 19 - Create Media Session

STEP PASSED

STEP 20 - Setup

STEP PASSED

STEP 21 - Create Sinks

STEP PASSED

STEP 22 - Play

STEP PASSED

STEP 23 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 24 - Teardown

STEP PASSED

STEP 25 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 26 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder, AudioEncoder, Metadata }]
STEP PASSED

STEP 27 - Checking the DUT returned single MediaProfile
STEP PASSED

STEP 28 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata
(MetadataConfigurationToken_1) }]
STEP PASSED

TEST PASSED

MEDIA2_RTSS-1-2-3-v21.06 MEDIA2 STREAMING – H.265 (RTP-Multicast, IPv4)

TestResult

STEP 1 - Get Device service address
STEP PASSED

STEP 2 - Check that the DUT returned Device service address
STEP PASSED

STEP 3 - Get Services
STEP PASSED

STEP 4 - Get Media2 service address
STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address
STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]
STEP PASSED

STEP 7 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1,
ProfileToken = ProfileToken_1]
STEP PASSED

STEP 8 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { AudioEncoder, Metadata }]
STEP PASSED

STEP 9 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 10 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { Metadata }]

STEP PASSED

STEP 11 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 12 - Get Stream Uri (Media2) [Protocol = RtspMulticast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 13 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 14 - Check if the stream uri has correct IP type

STEP PASSED

STEP 15 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 16 - Checking filters

STEP PASSED

STEP 17 - Describe

STEP PASSED

STEP 18 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 19 - Create Media Session

STEP PASSED

STEP 20 - Setup

STEP PASSED

STEP 21 - Create Sinks

STEP PASSED

STEP 22 - Play

STEP PASSED

STEP 23 - Waiting for 100 frames up to 10000 ms

STEP PASSED

STEP 24 - Teardown

STEP PASSED

STEP 25 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 26 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoSource, VideoEncoder, AudioEncoder, Metadata }]

STEP PASSED

STEP 27 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 28 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

TEST PASSED

MEDIA2_RTSS-4-1-1-v21.12 METADATA STREAMING (RTP-Unicast/UDP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Metadata Configuration Options (Media2) [ProfileToken = ProfileToken_1, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returns appropriate MetaData Options

STEP PASSED

STEP 9 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 10 - Get Stream Uri (Media2) [Protocol = RtspUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 11 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 12 - Check if the stream uri has correct IP type

STEP PASSED

STEP 13 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 21 - Waiting for 10 seconds

STEP PASSED

STEP 22 - Teardown

STEP PASSED

STEP 23 - Checking media frames count

STEP PASSED

STEP 24 - Set Metadata Configuration (Media2)

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 25 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 26 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-4-1-2-v20.12 METADATA STREAMING (RTP-Unicast/RTSP/HTTP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Metadata Configuration Options (Media2) [ProfileToken = ProfileToken_1, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returns appropriate MetaData Options

STEP PASSED

STEP 9 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 10 - Get Stream Uri (Media2) [Protocol = RtspOverHttp, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 11 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 12 - Check if the stream uri has correct IP type

STEP PASSED

STEP 13 - Check if the stream uri has the same port with the web service

STEP PASSED

STEP 14 - Check if the stream uri has the same scheme with the web service

STEP PASSED

STEP 15 - Describe

STEP PASSED

STEP 16 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 17 - Create Media Session

STEP PASSED

STEP 18 - Setup

STEP PASSED

STEP 19 - Create Sinks

STEP PASSED

STEP 20 - Play

STEP PASSED

STEP 21 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 22 - Waiting for 10 seconds

STEP PASSED

STEP 23 - Teardown

STEP PASSED

STEP 24 - Checking media frames count

STEP PASSED

STEP 25 - Set Metadata Configuration (Media2)

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 26 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 27 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-4-1-3-v20.12 METADATA STREAMING (RTP/RTSP/TCP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Metadata Configuration Options (Media2) [ProfileToken = ProfileToken_1, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returns appropriate MetaData Options

STEP PASSED

STEP 9 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 10 - Get Stream Uri (Media2) [Protocol = RTSP, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 11 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 12 - Check if the stream uri has correct IP type

STEP PASSED

STEP 13 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 21 - Waiting for 10 seconds

STEP PASSED

STEP 22 - Teardown

STEP PASSED

STEP 23 - Checking media frames count

STEP PASSED

STEP 24 - Set Metadata Configuration (Media2)

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 25 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 26 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-4-1-4-v21.12 METADATA STREAMING - SET SYNCHRONIZATION POINT

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Metadata Configuration Options (Media2) [ProfileToken = ProfileToken_1, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returns appropriate MetaData Options

STEP PASSED

STEP 9 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 10 - Get Stream Uri (Media2) [Protocol = RtpUnicast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 11 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 12 - Check if the stream uri has correct IP type

STEP PASSED

STEP 13 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 14 - Describe

STEP PASSED

STEP 15 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 16 - Create Media Session

STEP PASSED

STEP 17 - Setup

STEP PASSED

STEP 18 - Create Sinks

STEP PASSED

STEP 19 - Play

STEP PASSED

STEP 20 - Set Synchronization Point (Media2)

STEP PASSED

STEP 21 - Waiting for 10 seconds

STEP PASSED

STEP 22 - Teardown

STEP PASSED

STEP 23 - Checking media frames count

STEP PASSED

STEP 24 - Set Metadata Configuration (Media2)

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 25 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 26 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2_RTSS-4-2-1-v20.12 METADATA STREAMING (RTP-Multicast/UDP)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Get Metadata Configuration Options (Media2) [ProfileToken = ProfileToken_1, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returns appropriate MetaData Options

STEP PASSED

STEP 9 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 10 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { VideoEncoder, AudioEncoder }]

STEP PASSED

STEP 11 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { VideoEncoder }]

STEP PASSED

STEP 12 - Get Stream Uri (Media2) [Protocol = RtspMulticast, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 13 - Check if the stream uri is not longer than 128 octets

STEP PASSED

STEP 14 - Check if the stream uri has correct IP type

STEP PASSED

STEP 15 - Check if the stream uri has the scheme equal to 'rtsp'

STEP PASSED

STEP 16 - Describe

STEP PASSED

STEP 17 - Check of IP address type in response to RTSP DESCRIBE

STEP PASSED

STEP 18 - Create Media Session

STEP PASSED

STEP 19 - Setup

STEP PASSED

STEP 20 - Create Sinks

STEP PASSED

STEP 21 - Play

STEP PASSED

STEP 22 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 23 - Waiting for 10 seconds

STEP PASSED

STEP 24 - Teardown

STEP PASSED

STEP 25 - Checking media frames count

STEP PASSED

STEP 26 - Set Metadata Configuration (Media2)

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 27 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 28 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 29 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

TEST PASSED

Imaging

IMAGING-1-1-1-v17.12 IMAGING COMMAND GETIMAGINGSETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging settings

STEP PASSED

TEST PASSED

IMAGING-1-1-3-v19.12 IMAGING COMMAND GETOPTIONS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging options

STEP PASSED

STEP 7 - Check if the DUT sent imaging options

STEP PASSED

STEP 8 - Validate options structure

STEP PASSED

TEST PASSED

IMAGING-1-1-8-v19.12 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID SETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging options

STEP PASSED

STEP 7 - Check if the DUT sent imaging options

STEP PASSED

STEP 8 - Get imaging settings

STEP PASSED

STEP 9 - Check if the DUT sent imaging settings

STEP PASSED

STEP 10 - Validate options structure

STEP PASSED

STEP 11 - Set imaging settings

STEP PASSED

STEP 12 - Get imaging settings

STEP PASSED

STEP 13 - Check if the DUT sent imaging settings

STEP PASSED

STEP 14 - Check that settings have not been changed

STEP PASSED

TEST PASSED

**IMAGING-1-1-10-v17.12 IMAGING COMMAND GETIMAGINGSETTINGS – INVALID
VIDEOSOURCETOKEN**

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging settings - negative test

STEP PASSED

TEST PASSED

IMAGING-1-1-11-v17.12 IMAGING COMMAND GETOPTIONS – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get options - negative test

STEP PASSED

TEST PASSED

IMAGING-1-1-12-v17.12 IMAGING COMMAND SETIMAGINGSETTINGS – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Set imaging settings - negative test

STEP PASSED

TEST PASSED

IMAGING-1-1-14-v21.12 IMAGING COMMAND SETIMAGINGSETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging options

STEP PASSED

STEP 7 - Validate options structure

STEP PASSED

STEP 8 - Get imaging settings

STEP PASSED

STEP 9 - Set imaging settings

STEP PASSED

STEP 10 - Get imaging settings

STEP PASSED

STEP 11 - Check setting 'Backlight Compensation/Mode' is applied

STEP PASSED

STEP 12 - Set imaging settings

STEP PASSED

STEP 13 - Get imaging settings

STEP PASSED

STEP 14 - Check setting 'Backlight Compensation/Mode' is restored

STEP PASSED

STEP 15 - Set imaging settings

STEP PASSED

STEP 16 - Get imaging settings

STEP PASSED

STEP 17 - Check setting 'Backlight Compensation/Mode' is applied

STEP PASSED

STEP 18 - Set imaging settings

STEP PASSED

STEP 19 - Get imaging settings

STEP PASSED

STEP 20 - Check setting 'Backlight Compensation/Mode' is restored

STEP PASSED

STEP 21 - Set imaging settings

STEP PASSED

STEP 22 - Get imaging settings

STEP PASSED

STEP 23 - Check setting 'Brightness' is applied

STEP PASSED

STEP 24 - Set imaging settings

STEP PASSED

STEP 25 - Get imaging settings

STEP PASSED

STEP 26 - Check setting 'Brightness' is restored

STEP PASSED

STEP 27 - Set imaging settings

STEP PASSED

STEP 28 - Get imaging settings

STEP PASSED

STEP 29 - Check setting 'ColorSaturation' is applied

STEP PASSED

STEP 30 - Set imaging settings

STEP PASSED

STEP 31 - Get imaging settings

STEP PASSED

STEP 32 - Check setting 'ColorSaturation' is restored

STEP PASSED

STEP 33 - Set imaging settings

STEP PASSED

STEP 34 - Get imaging settings

STEP PASSED

STEP 35 - Check setting 'Contrast' is applied

STEP PASSED

STEP 36 - Set imaging settings

STEP PASSED

STEP 37 - Get imaging settings

STEP PASSED

STEP 38 - Check setting 'Contrast' is restored

STEP PASSED

STEP 39 - Set imaging settings

STEP PASSED

STEP 40 - Get imaging settings

STEP PASSED

STEP 41 - Check setting 'Sharpness' is applied

STEP PASSED

STEP 42 - Set imaging settings

STEP PASSED

STEP 43 - Get imaging settings

STEP PASSED

STEP 44 - Check setting 'Sharpness' is restored

STEP PASSED

STEP 45 - Set imaging settings

STEP PASSED

STEP 46 - Get imaging settings

STEP PASSED

STEP 47 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 48 - Set imaging settings

STEP PASSED

STEP 49 - Get imaging settings

STEP PASSED

STEP 50 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 51 - Set imaging settings

STEP PASSED

STEP 52 - Get imaging settings

STEP PASSED

STEP 53 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 54 - Set imaging settings

STEP PASSED

STEP 55 - Get imaging settings

STEP PASSED

STEP 56 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 57 - Set imaging settings

STEP PASSED

STEP 58 - Get imaging settings

STEP PASSED

STEP 59 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 60 - Set imaging settings

STEP PASSED

STEP 61 - Get imaging settings

STEP PASSED

STEP 62 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 63 - Set imaging settings

STEP PASSED

STEP 64 - Get imaging settings

STEP PASSED

STEP 65 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 66 - Set imaging settings

STEP PASSED

STEP 67 - Get imaging settings

STEP PASSED

STEP 68 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 69 - Set imaging settings

STEP PASSED

STEP 70 - Get imaging settings

STEP PASSED

STEP 71 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 72 - Set imaging settings

STEP PASSED

STEP 73 - Get imaging settings

STEP PASSED

STEP 74 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 75 - Set imaging settings

STEP PASSED

STEP 76 - Get imaging settings

STEP PASSED

STEP 77 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 78 - Set imaging settings

STEP PASSED

STEP 79 - Get imaging settings

STEP PASSED

STEP 80 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 81 - Set imaging settings

STEP PASSED

STEP 82 - Get imaging settings

STEP PASSED

STEP 83 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 84 - Set imaging settings

STEP PASSED

STEP 85 - Get imaging settings

STEP PASSED

STEP 86 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 87 - Set imaging settings

STEP PASSED

STEP 88 - Get imaging settings

STEP PASSED

STEP 89 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 90 - Set imaging settings

STEP PASSED

STEP 91 - Get imaging settings

STEP PASSED

STEP 92 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 93 - Set imaging settings

STEP PASSED

STEP 94 - Get imaging settings

STEP PASSED

STEP 95 - Check setting 'Exposure/Mode' is applied

STEP PASSED

STEP 96 - Set imaging settings

STEP PASSED

STEP 97 - Get imaging settings

STEP PASSED

STEP 98 - Check setting 'Exposure/Mode' is restored

STEP PASSED

STEP 99 - Set imaging settings

STEP PASSED

STEP 100 - Get imaging settings

STEP PASSED

STEP 101 - Check setting 'Focus/AutoFocusMode' is applied

STEP PASSED

STEP 102 - Set imaging settings

STEP PASSED

STEP 103 - Get imaging settings

STEP PASSED

STEP 104 - Check setting 'Focus/AutoFocusMode' is restored

STEP PASSED

STEP 105 - Set imaging settings

STEP PASSED

STEP 106 - Get imaging settings

STEP PASSED

STEP 107 - Check setting 'Focus/AutoFocusMode' is applied

STEP PASSED

STEP 108 - Set imaging settings

STEP PASSED

STEP 109 - Get imaging settings

STEP PASSED

STEP 110 - Check setting 'Focus/AutoFocusMode' is restored

STEP PASSED

STEP 111 - Set imaging settings

STEP PASSED

STEP 112 - Get imaging settings

STEP PASSED

STEP 113 - Check setting 'Focus/AutoFocusMode' is applied

STEP PASSED

STEP 114 - Set imaging settings

STEP PASSED

STEP 115 - Get imaging settings

STEP PASSED

STEP 116 - Check setting 'Focus/AutoFocusMode' is restored

STEP PASSED

STEP 117 - Set imaging settings

STEP PASSED

STEP 118 - Get imaging settings

STEP PASSED

STEP 119 - Check setting 'Focus/AutoFocusMode' is applied

STEP PASSED

STEP 120 - Set imaging settings

STEP PASSED

STEP 121 - Get imaging settings

STEP PASSED

STEP 122 - Check setting 'Focus/AutoFocusMode' is restored

STEP PASSED

STEP 123 - Set imaging settings

STEP PASSED

STEP 124 - Get imaging settings

STEP PASSED

STEP 125 - Check setting 'IrCutFilter/Mode' is applied

STEP PASSED

STEP 126 - Set imaging settings

STEP PASSED

STEP 127 - Get imaging settings

STEP PASSED

STEP 128 - Check setting 'IrCutFilter/Mode' is restored

STEP PASSED

STEP 129 - Set imaging settings

STEP PASSED

STEP 130 - Get imaging settings

STEP PASSED

STEP 131 - Check setting 'WhiteBalance/Mode' is applied

STEP PASSED

STEP 132 - Set imaging settings

STEP PASSED

STEP 133 - Get imaging settings

STEP PASSED

STEP 134 - Check setting 'WhiteBalance/Mode' is restored

STEP PASSED

STEP 135 - Set imaging settings

STEP PASSED

STEP 136 - Get imaging settings

STEP PASSED

STEP 137 - Check setting 'WhiteBalance/Mode' is applied

STEP PASSED

STEP 138 - Set imaging settings

STEP PASSED

STEP 139 - Get imaging settings

STEP PASSED

STEP 140 - Check setting 'WhiteBalance/Mode' is restored

STEP PASSED

STEP 141 - Set imaging settings

STEP PASSED

STEP 142 - Get imaging settings

STEP PASSED

STEP 143 - Check setting 'WhiteBalance/Mode' is applied

STEP PASSED

STEP 144 - Set imaging settings

STEP PASSED

STEP 145 - Get imaging settings

STEP PASSED

STEP 146 - Check setting 'WhiteBalance/Mode' is restored

STEP PASSED

STEP 147 - Set imaging settings

STEP PASSED

STEP 148 - Get imaging settings

STEP PASSED

STEP 149 - Check setting 'WideDynamicRange/Mode' is applied

STEP PASSED

STEP 150 - Set imaging settings

STEP PASSED

STEP 151 - Get imaging settings

STEP PASSED

STEP 152 - Check setting 'WideDynamicRange/Mode' is restored

STEP PASSED

STEP 153 - Set imaging settings

STEP PASSED

STEP 154 - Get imaging settings

STEP PASSED

STEP 155 - Check setting 'WideDynamicRange/Mode' is applied

STEP PASSED

STEP 156 - Set imaging settings

STEP PASSED

STEP 157 - Get imaging settings

STEP PASSED

STEP 158 - Check setting 'WideDynamicRange/Mode' is restored

STEP PASSED

TEST PASSED

IMAGING-1-1-15-v19.12 IMAGING COMMAND SET/IMAGING/SETTINGS ADDITIONAL FEATURES

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging options

STEP PASSED

STEP 7 - Validate options structure

STEP PASSED

STEP 8 - Get imaging settings

STEP PASSED

STEP 9 - Restore imaging settings

STEP PASSED

TEST PASSED

IMAGING-1-1-16-v19.12 GET IMAGING SETTINGS AND GET OPTIONS CONSISTENCY

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging settings

STEP PASSED

STEP 7 - Get imaging options

STEP PASSED

STEP 8 - Check if Imaging Settings contains Backlight Compensation Mode item is equal to one of Backlight Compensation Mode items in Imaging Options

STEP PASSED

STEP 9 - Check if Imaging Settings contains Backlight Compensation level item is greater than or equal to Backlight Compensation Mode Min item in Imaging Options

STEP PASSED

STEP 10 - Check if Imaging Settings contains Backlight Compensation level item is less than or equal to Backlight Compensation Mode Max item in Imaging Options

STEP PASSED

STEP 11 - Check if Imaging Settings contains Brightness item is greater than or equal to Brightness Min item in Imaging Options

STEP PASSED

STEP 12 - Check if Imaging Settings contains Brightness item is less than or equal to Brightness Max item in Imaging Options

STEP PASSED

STEP 13 - Check if Imaging Settings contains Color Saturation item is greater than or equal to Color Saturation Min item in Imaging Options

STEP PASSED

STEP 14 - Check if Imaging Settings contains Color Saturation item is less than or equal to Color Saturation Max item in Imaging Options

STEP PASSED

STEP 15 - Check if Imaging Settings contains Contrast item is greater than or equal to Contrast Min item in Imaging Options

STEP PASSED

STEP 16 - Check if Imaging Settings contains Contrast item is less than or equal to Contrast Max item in Imaging Options

STEP PASSED

STEP 17 - Check if Imaging Settings contains Exposure Mode item is equal to one of Exposure Mode items in Imaging Options

STEP PASSED

STEP 18 - Check if Imaging Settings contains Exposure Priority item is equal to one of Exposure Priority items in Imaging Options

STEP PASSED

STEP 19 - Check if Imaging Settings contains Exposure MinExposureTime item is greater than or equal to Exposure MinExposureTime Min item in Imaging Options

STEP PASSED

STEP 20 - Check if Imaging Settings contains Exposure MinExposureTime item is less than or equal to Exposure MinExposureTime Max item in Imaging Options

STEP PASSED

STEP 21 - Check if Imaging Settings contains Exposure MaxExposureTime item is greater than or equal to Exposure MaxExposureTime Min item in Imaging Options

STEP PASSED

STEP 22 - Check if Imaging Settings contains Exposure MaxExposureTime item is less than or equal to Exposure MaxExposureTime Max item in Imaging Options

STEP PASSED

STEP 23 - Check if Imaging Settings contains Exposure MinGain item is greater than or equal to Exposure MinGain Min item in Imaging Options

STEP PASSED

STEP 24 - Check if Imaging Settings contains Exposure MinGain item is less than or equal to Exposure MinGain Max item in Imaging Options

STEP PASSED

STEP 25 - Check if Imaging Settings contains Exposure MaxGain item is greater than or equal to Exposure MaxGain Min item in Imaging Options
STEP PASSED

STEP 26 - Check if Imaging Settings contains Exposure MaxGain item is less than or equal to Exposure MaxGain Max item in Imaging Options
STEP PASSED

STEP 27 - Check if Imaging Settings contains Exposure MinIris item is greater than or equal to Exposure MinIris Min item in Imaging Options
STEP PASSED

STEP 28 - Check if Imaging Settings contains Exposure MinIris item is less than or equal to Exposure MinIris Max item in Imaging Options
STEP PASSED

STEP 29 - Check if Imaging Settings contains Exposure MaxIris item is greater than or equal to Exposure MaxIris Min item in Imaging Options
STEP PASSED

STEP 30 - Check if Imaging Settings contains Exposure MaxIris item is less than or equal to Exposure MaxIris Max item in Imaging Options
STEP PASSED

STEP 31 - Check if Imaging Settings contains Exposure ExposureTime item is greater than or equal to Exposure ExposureTime Min item in Imaging Options
STEP PASSED

STEP 32 - Check if Imaging Settings contains Exposure ExposureTime item is less than or equal to Exposure ExposureTime Max item in Imaging Options
STEP PASSED

STEP 33 - Check if Imaging Settings contains Exposure Gain item is greater than or equal to Exposure Gain Min item in Imaging Options
STEP PASSED

STEP 34 - Check if Imaging Settings contains Exposure Gain item is less than or equal to Exposure Gain Max item in Imaging Options
STEP PASSED

STEP 35 - Check if Imaging Settings contains Exposure Iris item is greater than or equal to Exposure Iris Min item in Imaging

Options

STEP PASSED

STEP 36 - Check if Imaging Settings contains Exposure Iris item is less than or equal to Exposure Iris Max item in Imaging Options

STEP PASSED

STEP 37 - Check if Imaging Settings contains Auto Focus Mode item is equal to one of Auto Focus Mode items in Imaging Options

STEP PASSED

STEP 38 - Check if Imaging Settings contains Focus Default Speed item is greater than or equal to Focus Default Speed Min item in Imaging Options

STEP PASSED

STEP 39 - Check if Imaging Settings contains Focus Default Speed item is less than or equal to Focus Default Speed Max item in Imaging Options

STEP PASSED

STEP 40 - Check if Imaging Settings contains Focus Near Limit item is greater than or equal to Focus Near Limit Min item in Imaging Options

STEP PASSED

STEP 41 - Check if Imaging Settings contains Focus Near Limit item is less than or equal to Focus Near Limit Max item in Imaging Options

STEP PASSED

STEP 42 - Check if Imaging Settings contains Focus Far Limit item is greater than or equal to Focus Far Limit Min item in Imaging Options

STEP PASSED

STEP 43 - Check if Imaging Settings contains Focus Far Limit item is less than or equal to Focus Far Limit Max item in Imaging Options

STEP PASSED

STEP 44 - Check if Imaging Settings contains IrCut Filter item is equal to one of IrCut Filter Mode items in Imaging Options

STEP PASSED

STEP 45 - Check if Imaging Settings contains Sharpness item is greater than or equal to Sharpness Min item in Imaging Options

STEP PASSED

STEP 46 - Check if Imaging Settings contains Sharpness item is less than or equal to Sharpness Max item in Imaging Options

STEP PASSED

STEP 47 - Check if Imaging Settings contains Wide Dynamic Range Mode item is equal to one of Wide Dynamic Range Mode items in Imaging Options

STEP PASSED

STEP 48 - Check if Imaging Settings contains Wide Dynamic Range Level item is greater than or equal to Wide Dynamic Range Level Min item in Imaging Options

STEP PASSED

STEP 49 - Check if Imaging Settings contains Wide Dynamic Range Level item is less than or equal to Wide Dynamic Range Level Max item in Imaging Options

STEP PASSED

STEP 50 - Check if Imaging Settings contains White Balance Mode item is equal to one of White Balance Mode items in Imaging Options

STEP PASSED

STEP 51 - Check if Imaging Settings contains White Balance CrGain item is greater than or equal to White Balance YrGain Min item in Imaging Options

STEP PASSED

STEP 52 - Check if Imaging Settings contains White Balance CrGain item is less than or equal to White Balance YrGain Max item in Imaging Options

STEP PASSED

STEP 53 - Check if Imaging Settings contains White Balance CbGain item is greater than or equal to White Balance YbGain Min item in Imaging Options

STEP PASSED

STEP 54 - Check if Imaging Settings contains White Balance CbGain item is less than or equal to White Balance YbGain Max item in Imaging Options

STEP PASSED

TEST PASSED

IMAGING-2-1-1-v17.12 IMAGING COMMAND GETMOVEOPTIONS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

TEST PASSED

IMAGING-2-1-3-v17.12 IMAGING COMMAND ABSOLUTE MOVE

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Absolute Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

TEST PASSED

IMAGING-2-1-4-v17.12 IMAGING COMMAND ABSOLUTE MOVE – INVALID SETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Absolute Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

TEST PASSED

IMAGING-2-1-5-v17.12 IMAGING COMMAND RELATIVE MOVE

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Relative Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

TEST PASSED

IMAGING-2-1-6-v17.12 IMAGING COMMAND RELATIVE MOVE – INVALID SETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Relative Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

TEST PASSED

IMAGING-2-1-7-v17.12 IMAGING COMMAND CONTINUOUS MOVE

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Continuous Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

STEP 9 - Send Move command (VideoSourceToken_1)

STEP PASSED

STEP 10 - Stop

STEP PASSED

TEST PASSED

IMAGING-2-1-8-v17.12 IMAGING COMMAND CONTINUOUS MOVE – INVALID SETTINGS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Continuous Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

STEP 9 - Move - negative test (invalid Speed)

STEP PASSED

TEST PASSED

IMAGING-2-1-10-v17.12 IMAGING COMMAND MOVE – UNSUPPORTED MOVE

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get Move options for VideoSourceToken_1

STEP PASSED

STEP 7 - Validate Move options

STEP PASSED

STEP 8 - Check if Absolute Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

STEP 9 - Move - negative test (absolute not supported)

STEP PASSED

STEP 10 - Check if Relative Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

STEP 11 - Move - negative test (relative not supported)

STEP PASSED

STEP 12 - Check if Continuous Move is supported for video source 'VideoSourceToken_1'

STEP PASSED

TEST PASSED

IMAGING-2-1-11-v17.12 IMAGING COMMAND GETSTATUS

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get imaging status

STEP PASSED

TEST PASSED

IMAGING-2-1-13-v17.12 IMAGING COMMAND STOP

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Stop

STEP PASSED

TEST PASSED

IMAGING-2-1-15-v17.12 IMAGING COMMAND GETMOVEOPTIONS – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Get options - negative test

STEP PASSED

TEST PASSED

IMAGING-2-1-16-v17.12 IMAGING COMMAND MOVE – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Move - negative test

STEP PASSED

TEST PASSED

IMAGING-2-1-17-v17.12 IMAGING COMMAND GETSTATUS – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - GetStatus - negative test

STEP PASSED

TEST PASSED

IMAGING-2-1-18-v17.12 IMAGING COMMAND STOP – INVALID VIDEOSOURCETOKEN

TestResult

STEP 1 - Get imaging service address

STEP PASSED

STEP 2 - Get media service address

STEP PASSED

STEP 3 - Get I/O service address

STEP PASSED

STEP 4 - Get video sources

STEP PASSED

STEP 5 - Check that the DUT returned Video Sources

STEP PASSED

STEP 6 - Stop - negative test

STEP PASSED

TEST PASSED

IMAGING-3-1-1-v14.12 IMAGING SERVICE CAPABILITIES

TestResult

STEP 1 - Get Imaging service address

STEP PASSED

STEP 2 - Check that the DUT returned Imaging service address

STEP PASSED

STEP 3 - Get Service Capabilities

STEP PASSED

TEST PASSED

IMAGING-3-1-2-v14.12 GET SERVICES AND GET IMAGING SERVICE CAPABILITIES CONSISTENCY

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that the DUT returned Imaging service information

STEP PASSED

STEP 3 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 4 - Get Imaging service address

STEP PASSED

STEP 5 - Check that the DUT returned Imaging service address

STEP PASSED

STEP 6 - Get Service Capabilities

STEP PASSED

STEP 7 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 8 - Compare Capabilities

STEP PASSED

TEST PASSED

IMAGING-4-1-1-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO BLURRY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event Properties

STEP PASSED

STEP 5 - Check that the DUT returned at least one of the specified topics

STEP PASSED

STEP 6 - Checking description of event with topic tns1:VideoSource/ImageTooBlurry/ImagingService

STEP PASSED

STEP 7 - Create Pull Point Subscription

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Send PullMessages request

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Waiting for notifications

STEP PASSED

STEP 19 - Send Unsubscribe request

STEP PASSED

TEST PASSED

IMAGING-4-1-2-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO DARK

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event Properties

STEP PASSED

STEP 5 - Check that the DUT returned at least one of the specified topics

STEP PASSED

STEP 6 - Checking description of event with topic tns1:VideoSource/ImageTooDark/ImagingService

STEP PASSED

STEP 7 - Create Pull Point Subscription

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Send PullMessages request

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Waiting for notifications

STEP PASSED

STEP 19 - Send Unsubscribe request

STEP PASSED

TEST PASSED

IMAGING-4-1-3-v18.06 REALTIME PULLPOINT SUBSCRIPTION – IMAGE TOO BRIGHT

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event Properties

STEP PASSED

STEP 5 - Check that the DUT returned at least one of the specified topics

STEP PASSED

STEP 6 - Checking description of event with topic tns1:VideoSource/ImageTooBright/ImagingService

STEP PASSED

STEP 7 - Create Pull Point Subscription

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Send PullMessages request

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Waiting for notifications

STEP PASSED

STEP 19 - Send Unsubscribe request

STEP PASSED

TEST PASSED

IMAGING-4-1-4-v18.06 REALTIME PULLPOINT SUBSCRIPTION – GLOBAL SCENE CHANGE

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event Properties

STEP PASSED

STEP 5 - Check that the DUT returned at least one of the specified topics

STEP PASSED

STEP 6 - Checking description of event with topic tns1:VideoSource/GlobalSceneChange/ImagingService

STEP PASSED

STEP 7 - Create Pull Point Subscription

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Send PullMessages request

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Waiting for notifications

STEP PASSED

STEP 19 - Send Unsubscribe request

STEP PASSED

TEST PASSED

IMAGING-4-1-5-v18.06 REALTIME PULLPOINT SUBSCRIPTION – MOTION ALARM

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event Properties

STEP PASSED

STEP 5 - Check that the DUT returned at least one of the specified topics

STEP PASSED

STEP 6 - Checking description of event with topic tns1:VideoSource/MotionAlarm

STEP PASSED

STEP 7 - Create Pull Point Subscription

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Send PullMessages request

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Waiting for notifications

STEP PASSED

STEP 19 - Send Unsubscribe request

STEP PASSED

TEST PASSED

Device I/O

DEVICEIO-1-1-1-v16.07 IO GETRELAYOUTPUTS

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get relay outputs

STEP PASSED

STEP 3 - Check that the DUT sent relay outputs information

STEP PASSED

TEST PASSED

DEVICEIO-1-1-2-v17.12 IO GETRELAYOUTPUTS – VERIFY QUANTITY

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get Service Capabilities(Device I/O)

STEP PASSED

STEP 3 - Check that DUT returned capabilities

STEP PASSED

STEP 4 - Get relay outputs

STEP PASSED

STEP 5 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 6 - Check that count of relay outputs is the same

STEP PASSED

TEST PASSED

DEVICEIO-1-1-3-v16.07 IO GETRELAYOUTPUTOPTIONS

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get relay outputs

STEP PASSED

STEP 3 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 4 - Send GetRelayOutputOptions request

STEP PASSED

STEP 5 - Check response

STEP PASSED

TEST PASSED

DEVICEIO-1-1-4-v18.06SR1 IO SETRELAYOUTPUTSETTINGS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Device I/O service address

STEP PASSED

STEP 5 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 6 - Get Relay Outputs

STEP PASSED

STEP 7 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 8 - Get Relay Output Options (DeviceIO) [Token = RelayOutputToken_1]

STEP PASSED

STEP 9 - Check that the DUT sent only one relay output options item

STEP PASSED

STEP 10 - Check that the DUT sent relay output options item with 'RelayOutputToken_1' token

STEP PASSED

STEP 11 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]

STEP PASSED

STEP 12 - Get Relay Outputs

STEP PASSED

STEP 13 - Check that the DUT sent relay output item with 'RelayOutputToken_1' token

STEP PASSED

STEP 14 - Check if relay output item Mode = 'Bistable'

STEP PASSED

STEP 15 - Check if relay output item IdleState = 'open'

STEP PASSED

STEP 16 - Check that the DUT sent relay output options item with non empty DelayTimes field

STEP PASSED

STEP 17 - Check if DelayTimes field contains two values

STEP PASSED

STEP 18 - Check if the first delay time value is less then or equal to the second delay time value

STEP PASSED

STEP 19 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Monostable, DelayTime = PT5S, IdleState = closed]

STEP PASSED

STEP 20 - Get Relay Outputs

STEP PASSED

STEP 21 - Check that the DUT sent relay output item with 'RelayOutputToken_1' token

STEP PASSED

STEP 22 - Check if relay output item Mode = 'Monostable'

STEP PASSED

STEP 23 - Check if relay output item IdleState = 'closed'

STEP PASSED

STEP 24 - Check if relay output item DelayTime = 'PT5S'

STEP PASSED

STEP 25 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]

STEP PASSED

TEST PASSED

DEVICEIO-1-1-5-v16.07 IO SETRELAYOUTPUTSETTINGS – INVALID TOKEN

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get relay outputs

STEP PASSED

STEP 3 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 4 - Set relay output settings - negative test

STEP PASSED

TEST PASSED

DEVICEIO-1-2-1-v18.06SR1 IO SETRELAYOUTPUTSTATE – BISTABLE MODE (OPENED IDLE STATE)

TestResult

STEP 1 - Check that the DUT sent relay output with RelayMode = Bistable and RelayIdleState = open

STEP PASSED

STEP 2 - Get Device service address

STEP PASSED

STEP 3 - Check that the DUT returned Device service address

STEP PASSED

STEP 4 - Get Services

STEP PASSED

STEP 5 - Get Device I/O service address

STEP PASSED

STEP 6 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 7 - Get Relay Output Options (DeviceIO) [Token = RelayOutputToken_1]

STEP PASSED

STEP 8 - Check that the DUT sent relay output options

STEP PASSED

STEP 9 - Get Relay Outputs

STEP PASSED

STEP 10 - Check that the DUT sent relay output with token = RelayOutputToken_1

STEP PASSED

STEP 11 - Create Pull Point Subscription

STEP PASSED

STEP 12 - Check that TerminationTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Send PullMessages request

STEP PASSED

STEP 19 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 20 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 21 - Response is not empty

STEP PASSED

STEP 22 - Waiting for notification with PropertyOperation = 'Initialized'

STEP PASSED

STEP 23 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 24 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]
STEP PASSED

STEP 25 - Set Relay Output State
STEP PASSED

STEP 26 - Send PullMessages request
STEP PASSED

STEP 27 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 28 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse
STEP PASSED

STEP 29 - Response is not empty
STEP PASSED

STEP 30 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'active'
STEP PASSED

STEP 31 - Set Relay Output State
STEP PASSED

STEP 32 - Send PullMessages request
STEP PASSED

STEP 33 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 34 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse
STEP PASSED

STEP 35 - Response is not empty
STEP PASSED

STEP 36 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'inactive'
STEP PASSED

STEP 37 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]
STEP PASSED

STEP 38 - Send Unsubscribe request
STEP PASSED

TEST PASSED

DEVICEIO-1-2-2-v18.06SR1 IO SETRELAYOUTPUTSTATE – BISTABLE MODE (CLOSED IDLE STATE)

TestResult

STEP 1 - Check that the DUT sent relay output with RelayMode = Bistable and RelayIdleState = closed
STEP PASSED

STEP 2 - Get Device service address
STEP PASSED

STEP 3 - Check that the DUT returned Device service address
STEP PASSED

STEP 4 - Get Services
STEP PASSED

STEP 5 - Get Device I/O service address
STEP PASSED

STEP 6 - Check that the DUT returned Device I/O service address
STEP PASSED

STEP 7 - Get Relay Output Options (DeviceIO) [Token = RelayOutputToken_1]
STEP PASSED

STEP 8 - Check that the DUT sent relay output options
STEP PASSED

STEP 9 - Get Relay Outputs
STEP PASSED

STEP 10 - Check that the DUT sent relay output with token = RelayOutputToken_1

STEP PASSED

STEP 11 - Create Pull Point Subscription

STEP PASSED

STEP 12 - Check that TerminationTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Send PullMessages request

STEP PASSED

STEP 19 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 20 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 21 - Response is not empty

STEP PASSED

STEP 22 - Waiting for notification with PropertyOperation = 'Initialized'

STEP PASSED

STEP 23 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 24 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = closed]
STEP PASSED

STEP 25 - Set Relay Output State
STEP PASSED

STEP 26 - Send PullMessages request
STEP PASSED

STEP 27 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 28 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse
STEP PASSED

STEP 29 - Response is not empty
STEP PASSED

STEP 30 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'active'
STEP PASSED

STEP 31 - Set Relay Output State
STEP PASSED

STEP 32 - Send PullMessages request
STEP PASSED

STEP 33 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 34 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse
STEP PASSED

STEP 35 - Response is not empty
STEP PASSED

STEP 36 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'inactive'
STEP PASSED

STEP 37 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]
STEP PASSED

STEP 38 - Send Unsubscribe request
STEP PASSED

TEST PASSED

DEVICEIO-1-2-3-v19.12 IO SETRELAYOUTPUTSTATE – MONOSTABLE MODE (OPENED IDLE STATE)

TestResult

STEP 1 - Check that the DUT sent relay output with RelayMode = Monostable and RelayIdleState = open
STEP PASSED

STEP 2 - Get Device service address
STEP PASSED

STEP 3 - Check that the DUT returned Device service address
STEP PASSED

STEP 4 - Get Services
STEP PASSED

STEP 5 - Get Device I/O service address
STEP PASSED

STEP 6 - Check that the DUT returned Device I/O service address
STEP PASSED

STEP 7 - Get Relay Output Options (DeviceIO) [Token = RelayOutputToken_1]
STEP PASSED

STEP 8 - Check that the DUT sent relay output options
STEP PASSED

STEP 9 - Get Relay Outputs
STEP PASSED

STEP 10 - Check that the DUT sent relay output with token = RelayOutputToken_1

STEP PASSED

STEP 11 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Monostable, DelayTime = PT5S, IdleState = open]

STEP PASSED

STEP 12 - Create Pull Point Subscription

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 17 - Check if SubscriptionReference contains address

STEP PASSED

STEP 18 - Check that URL specified is valid

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notification with PropertyOperation = 'Initialized', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'active' or 'inactive'

STEP PASSED

STEP 24 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 25 - Send Unsubscribe request

STEP PASSED

STEP 26 - Create Pull Point Subscription

STEP PASSED

STEP 27 - Check that TerminationTime is specified

STEP PASSED

STEP 28 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 29 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 30 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 31 - Check if SubscriptionReference contains address

STEP PASSED

STEP 32 - Check that URL specified is valid

STEP PASSED

STEP 33 - Send PullMessages request

STEP PASSED

STEP 34 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 35 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 36 - Response is not empty

STEP PASSED

STEP 37 - Waiting for notification with PropertyOperation = 'Initialized', 'RelayToken' value = 'RelayOutputToken_1' and

'LogicalState' value = 'inactive'

STEP PASSED

STEP 38 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 39 - Set Relay Output State

STEP PASSED

STEP 40 - Send PullMessages request

STEP PASSED

STEP 41 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 42 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 43 - Response is not empty

STEP PASSED

STEP 44 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and

'LogicalState' value = 'active'

STEP PASSED

STEP 45 - Waiting 5 seconds ...

STEP PASSED

STEP 46 - Send PullMessages request

STEP PASSED

STEP 47 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 48 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 49 - Response is not empty

STEP PASSED

STEP 50 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and

'LogicalState' value = 'inactive'

STEP PASSED

STEP 51 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]

STEP PASSED

STEP 52 - Send Unsubscribe request

STEP PASSED

TEST PASSED

DEVICEIO-1-2-4-v19.12 IO SETRELAYOUTPUTSTATE – MONOSTABLE MODE (CLOSED IDLE STATE)

TestResult

STEP 1 - Check that the DUT sent relay output with RelayMode = Monostable and RelayIdleState = closed

STEP PASSED

STEP 2 - Get Device service address

STEP PASSED

STEP 3 - Check that the DUT returned Device service address

STEP PASSED

STEP 4 - Get Services

STEP PASSED

STEP 5 - Get Device I/O service address

STEP PASSED

STEP 6 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 7 - Get Relay Output Options (DeviceIO) [Token = RelayOutputToken_1]

STEP PASSED

STEP 8 - Check that the DUT sent relay output options

STEP PASSED

STEP 9 - Get Relay Outputs

STEP PASSED

STEP 10 - Check that the DUT sent relay output with token = RelayOutputToken_1

STEP PASSED

STEP 11 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Monostable, DelayTime = PT5S, IdleState = closed]

STEP PASSED

STEP 12 - Create Pull Point Subscription

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 17 - Check if SubscriptionReference contains address

STEP PASSED

STEP 18 - Check that URL specified is valid

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notification with PropertyOperation = 'Initialized', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'active' or 'inactive'

STEP PASSED

STEP 24 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 25 - Send Unsubscribe request

STEP PASSED

STEP 26 - Create Pull Point Subscription

STEP PASSED

STEP 27 - Check that TerminationTime is specified

STEP PASSED

STEP 28 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 29 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 30 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 31 - Check if SubscriptionReference contains address

STEP PASSED

STEP 32 - Check that URL specified is valid

STEP PASSED

STEP 33 - Send PullMessages request

STEP PASSED

STEP 34 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 35 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 36 - Response is not empty

STEP PASSED

STEP 37 - Waiting for notification with PropertyOperation = 'Initialized', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'inactive'

STEP PASSED

STEP 38 - Check the DUT sent notification with LogicalState

STEP PASSED

STEP 39 - Set Relay Output State

STEP PASSED

STEP 40 - Send PullMessages request

STEP PASSED

STEP 41 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 42 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 43 - Response is not empty

STEP PASSED

STEP 44 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'active'

STEP PASSED

STEP 45 - Waiting 5 seconds ...

STEP PASSED

STEP 46 - Send PullMessages request

STEP PASSED

STEP 47 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 48 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 49 - Response is not empty

STEP PASSED

STEP 50 - Waiting for notification with PropertyOperation = 'Changed', 'RelayToken' value = 'RelayOutputToken_1' and 'LogicalState' value = 'inactive'

STEP PASSED

STEP 51 - Set Relay Output Settings (DeviceIO) [Token = RelayOutputToken_1, Mode = Bistable, DelayTime = PT30S, IdleState = open]

STEP PASSED

STEP 52 - Send Unsubscribe request

STEP PASSED

TEST PASSED

DEVICEIO-2-1-1-v18.06 REALTIME PULLPOINT SUBSCRIPTION – DIGITAL INPUT EVENT

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

STEP 3 - Check that event with topic tns1:Device/Trigger/DigitalInput is present

STEP PASSED

STEP 4 - Checking description of event with topic tns1:Device/Trigger/DigitalInput

STEP PASSED

STEP 5 - Create Pull Point Subscription

STEP PASSED

STEP 6 - Check that TerminationTime is specified

STEP PASSED

STEP 7 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 8 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 9 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 10 - Check if SubscriptionReference contains address

STEP PASSED

STEP 11 - Check that URL specified is valid

STEP PASSED

STEP 12 - Send PullMessages request

STEP PASSED

STEP 13 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 14 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 15 - Response is not empty

STEP PASSED

STEP 16 - Waiting for notifications [Topic = 'tns1:Device/Trigger/DigitalInput', PropertyOperation = 'Initialized']

STEP PASSED

STEP 17 - Send Unsubscribe request

STEP PASSED

TEST PASSED

DEVICEIO-2-1-2-v17.12 DEVICE IO SERVICE TRIGGER EVENT CHECK

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

STEP 3 - Check that event with topic tns1:Device/Trigger/Relay is present

STEP PASSED

STEP 4 - Checking RelayToken type

STEP PASSED

TEST PASSED

DEVICEIO-3-1-1-v17.01 GETDIGITALINPUTS

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get Digital Inputs

STEP PASSED

STEP 3 - Check the DUT return at least one DigitalInput item

STEP PASSED

TEST PASSED

DEVICEIO-3-1-2-v17.01 GETDIGITALINPUTS – VERIFY QUANTITY

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get Service Capabilities(Device I/O)

STEP PASSED

STEP 3 - Get Digital Inputs

STEP PASSED

STEP 4 - Check the DUT return at least one DigitalInput item

STEP PASSED

STEP 5 - Check that the number of items returned in the 'GetDigitalInputsResponse' by the DUT is the same as specified in Device I/O ServiceCapabilities.DigitalInputs

STEP PASSED

TEST PASSED

DEVICEIO-3-1-3-v17.12 I/O GET DIGITAL INPUT CONFIGURATION OPTIONS

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get Digital Inputs

STEP PASSED

STEP 3 - Check the DUT return at least one DigitalInput item

STEP PASSED

STEP 4 - Get Digital Input Configuration Options

STEP PASSED

STEP 5 - Get Digital Input Configuration Options

STEP PASSED

TEST PASSED

DEVICEIO-3-1-4-v17.12 I/O DIGITAL INPUT CONFIGURATION

TestResult

STEP 1 - Get Device IO service address

STEP PASSED

STEP 2 - Get Digital Inputs

STEP PASSED

STEP 3 - Check the DUT return at least one DigitalInput item

STEP PASSED

STEP 4 - Get Digital Input Configuration Options

STEP PASSED

STEP 5 - Set Digital Input Configurations

STEP PASSED

STEP 6 - Get Digital Inputs

STEP PASSED

STEP 7 - Check the DUT return at least one DigitalInput item

STEP PASSED

STEP 8 - Check the DUT successfully changed value of 'IdleState' field

STEP PASSED

STEP 9 - Set Digital Input Configurations

STEP PASSED

STEP 10 - Get Digital Inputs

STEP PASSED

STEP 11 - Check the DUT return at least one DigitalInput item

STEP PASSED

STEP 12 - Check the DUT successfully changed value of 'IdleState' field

STEP PASSED

TEST PASSED

DEVICEIO-7-1-1-v17.12 IO GET VIDEO SOURCES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Device I/O service address

STEP PASSED

STEP 5 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 6 - Get Service Capabilities

STEP PASSED

STEP 7 - Get Video Sources

STEP PASSED

STEP 8 - Check if the DUT returned at least one VideoSource item

STEP PASSED

STEP 9 - Check if the number of VideoSource items is equal to VideoSources value in IOServiceCapabilities item

STEP PASSED

STEP 10 - Check if the DUT did not return VideoSource items with the same token

STEP PASSED

TEST PASSED

Media 2 Configuration

MEDIA2-1-1-1-v17.06 READY TO USE MEDIA PROFILE FOR VIDEO STREAMING

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Device I/O service address

STEP PASSED

STEP 5 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 6 - Get Video Sources

STEP PASSED

STEP 7 - Checking the DUT returned at least one VideoSource item

STEP PASSED

STEP 8 - Get Media2 service address

STEP PASSED

STEP 9 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 10 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 11 - Check the DUT returns appropriate Media Profile

STEP PASSED

TEST PASSED

MEDIA2-1-1-2-v20.06 CREATE MEDIA PROFILE WITH PRE-DEFINED CONFIGURATION

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Create Pull Point Subscription

STEP PASSED

STEP 11 - Check that TerminationTime is specified

STEP PASSED

STEP 12 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 13 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 14 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 15 - Check if SubscriptionReference contains address

STEP PASSED

STEP 16 - Check that URL specified is valid

STEP PASSED

STEP 17 - Create Profile (Media2) [Name = testMedia2, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 18 - Send PullMessages request

STEP PASSED

STEP 19 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 20 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 21 - Response is not empty

STEP PASSED

STEP 22 - Waiting for notification

STEP PASSED

STEP 23 - Get Profiles (Media2) [Token = ProfileToken_9, Type = { VideoSource }]

STEP PASSED

STEP 24 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 25 - Checking value of 'token' field of received MediaProfile item

STEP PASSED

STEP 26 - Checking consistency of 'CreateProfile' and 'GetProfiles' commands

STEP PASSED

STEP 27 - Delete Profile (Media2) [Token = ProfileToken_9]

STEP PASSED

STEP 28 - Send PullMessages request

STEP PASSED

STEP 29 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 30 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 31 - Response is not empty

STEP PASSED

STEP 32 - Waiting for notification

STEP PASSED

STEP 33 - Get Profiles (Media2) [Token = ProfileToken_9, Type = { }]

STEP PASSED

STEP 34 - Send Unsubscribe request

STEP PASSED

TEST PASSED

MEDIA2-1-1-3-v20.12 DYNAMIC MEDIA PROFILE CONFIGURATION

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 8 - Create Profile (Media2) [Name = testMedia2, no Configuration]

STEP PASSED

STEP 9 - Create Pull Point Subscription

STEP PASSED

STEP 10 - Check that TerminationTime is specified

STEP PASSED

STEP 11 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 12 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 13 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 14 - Check if SubscriptionReference contains address

STEP PASSED

STEP 15 - Check that URL specified is valid

STEP PASSED

STEP 16 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 17 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 18 - Add Configuration (Media2) [ProfileToken = ProfileToken_10, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notification

STEP PASSED

STEP 24 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource }]

STEP PASSED

STEP 25 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 26 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 27 - Check the DUT returned MediaProfile item with valid Video Source configuration

STEP PASSED

STEP 28 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_10]

STEP PASSED

STEP 29 - Check the DUT returned Video Encoder configuration

STEP PASSED

STEP 30 - Add Configuration (Media2) [ProfileToken = ProfileToken_10, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

STEP 31 - Send PullMessages request

STEP PASSED

STEP 32 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 33 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 34 - Response is not empty

STEP PASSED

STEP 35 - Waiting for notification

STEP PASSED

STEP 36 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 37 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 38 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 39 - Check the DUT returned MediaProfile item with valid Video Source configuration

STEP PASSED

STEP 40 - Check the DUT returned MediaProfile item with valid Video Encoder configuration

STEP PASSED

STEP 41 - Remove Configuration (Media2) [ProfileToken = ProfileToken_10, Configuration = { VideoEncoder }]

STEP PASSED

STEP 42 - Send PullMessages request

STEP PASSED

STEP 43 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 44 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 45 - Response is not empty

STEP PASSED

STEP 46 - Waiting for notification

STEP PASSED

STEP 47 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 48 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 49 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 50 - Check the DUT returned MediaProfile item with valid Video Source configuration

STEP PASSED

STEP 51 - Check the DUT returned MediaProfile item without Video Encoder configuration

STEP PASSED

STEP 52 - Get Metadata Configurations (Media2) [ProfileToken = ProfileToken_10, no ConfigurationToken]

STEP PASSED

STEP 53 - Add Configuration (Media2) [ProfileToken = ProfileToken_10, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 54 - Send PullMessages request

STEP PASSED

STEP 55 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 56 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 57 - Response is not empty

STEP PASSED

STEP 58 - Waiting for notification

STEP PASSED

STEP 59 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource, Metadata }]

STEP PASSED

STEP 60 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 61 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 62 - Check the DUT returned MediaProfile item with valid Video Source configuration

STEP PASSED

STEP 63 - Check the DUT returned MediaProfile item with valid Metadata configuration

STEP PASSED

STEP 64 - Remove Configuration (Media2) [ProfileToken = ProfileToken_10, Configuration = { Metadata }]

STEP PASSED

STEP 65 - Send PullMessages request

STEP PASSED

STEP 66 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 67 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 68 - Response is not empty

STEP PASSED

STEP 69 - Waiting for notification

STEP PASSED

STEP 70 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource, Metadata }]

STEP PASSED

STEP 71 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 72 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 73 - Check the DUT returned MediaProfile item with valid Video Source configuration

STEP PASSED

STEP 74 - Check the DUT returned MediaProfile item without Metadata configuration

STEP PASSED

STEP 75 - Remove Configuration (Media2) [ProfileToken = ProfileToken_10, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 76 - Send PullMessages request

STEP PASSED

STEP 77 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 78 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 79 - Response is not empty

STEP PASSED

STEP 80 - Waiting for notification

STEP PASSED

STEP 81 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { VideoSource }]

STEP PASSED

STEP 82 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 83 - Check the DUT returned MediaProfile item with valid token

STEP PASSED

STEP 84 - Check the DUT returned MediaProfile item without Video Source configuration

STEP PASSED

STEP 85 - Check the DUT has returned at least one non empty Analytics Configuration list (if supported)

STEP PASSED

STEP 86 - Delete Profile (Media2) [Token = ProfileToken_10]

STEP PASSED

STEP 87 - Get Profiles (Media2) [Token = ProfileToken_10, Type = { }]

STEP PASSED

STEP 88 - Send Unsubscribe request

STEP PASSED

TEST PASSED

MEDIA2-1-1-4-v19.12 GET PROFILES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Check the DUT returned 1st list of MediaProfiles with the different tokens

STEP PASSED

STEP 8 - Get Profiles (Media2) [no Token, no Type]

STEP PASSED

STEP 9 - Check the DUT returned 2nd list of MediaProfiles with the different tokens

STEP PASSED

STEP 10 - Check the DUT returned 1st and 2nd lists of MediaProfiles where number of profiles are equal

STEP PASSED

STEP 11 - Check the DUT returned MediaProfile with the same token as in 1st list

STEP PASSED

STEP 12 - Check the DUT returned MediaProfile with empty Configurations element

STEP PASSED

STEP 13 - Check the DUT returned MediaProfile with the same token as in 1st list

STEP PASSED

STEP 14 - Check the DUT returned MediaProfile with empty Configurations element

STEP PASSED

STEP 15 - Get Profiles (Media2) [no Token, Type = { VideoSource }]

STEP PASSED

STEP 16 - Check the DUT returned 3rd list of MediaProfiles with the different tokens

STEP PASSED

STEP 17 - Check the DUT returned 1st and 3rd lists of MediaProfiles where number of profiles are equal

STEP PASSED

STEP 18 - Check the DUT returned list of MediaProfiles without profile with extra Configurations

STEP PASSED

STEP 19 - Check the DUT returned MediaProfile with different VideoSource

STEP PASSED

STEP 20 - Compare VideoSourceConfiguration of MediaProfile in 1st list and VideoSourceConfiguration of MediaProfile in 3rd

STEP PASSED

STEP 21 - Check the DUT returned list of MediaProfiles without profile with extra Configurations

STEP PASSED

STEP 22 - Check the DUT returned MediaProfile with different VideoSource

STEP PASSED

STEP 23 - Compare VideoSourceConfiguration of MediaProfile in 1st list and VideoSourceConfiguration of MediaProfile in 3rd

STEP PASSED

STEP 24 - Get Profiles (Media2) [no Token, Type = { Metadata }]

STEP PASSED

STEP 25 - Check the DUT returned 5th list of MediaProfiles with the different tokens

STEP PASSED

STEP 26 - Check the DUT returned 1st and 5th lists of MediaProfiles where number of profiles are equal

STEP PASSED

STEP 27 - Check the DUT returned list of MediaProfiles without profile with extra Configurations

STEP PASSED

STEP 28 - Check the DUT returned MediaProfile with different Metadata configuration

STEP PASSED

STEP 29 - Compare MetadataConfiguration of MediaProfile in 1st list and MetadataConfiguration of MediaProfile in 5th

STEP PASSED

STEP 30 - Check the DUT returned list of MediaProfiles without profile with extra Configurations

STEP PASSED

STEP 31 - Check the DUT returned MediaProfile with different Metadata configuration

STEP PASSED

STEP 32 - Compare MetadataConfiguration of MediaProfile in 1st list and MetadataConfiguration of MediaProfile in 5th

STEP PASSED

TEST PASSED

MEDIA2-1-1-5-v20.12 CREATE MEDIA PROFILE WITH CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Create Profile (Media2) [Name = testMedia, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 11 - Get Profiles (Media2) [Token = ProfileToken_11, Type = { All }]

STEP PASSED

STEP 12 - Check that list of MediaProfiles is not empty

STEP PASSED

STEP 13 - Check that list of MediaProfiles contains exactly one item

STEP PASSED

STEP 14 - Checking value of 'token' field of received MediaProfile item

STEP PASSED

STEP 15 - Checking consistency of 'CreateProfile' and 'GetProfiles' commands

STEP PASSED

STEP 16 - Delete Profile (Media2) [Token = ProfileToken_11]

STEP PASSED

TEST PASSED

MEDIA2-1-1-6-v20.06 REMOVE ALL CONFIGURATIONS FROM MEDIA PROFILE

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { All }]

STEP PASSED

STEP 8 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 9 - Check if GetProfiles returned at least one MediaProfile item

STEP PASSED

STEP 10 - Check if MediaProfile does not contain configurations

STEP PASSED

STEP 11 - Remove Configuration (Media2) [ProfileToken = ProfileToken_2, Configuration = { All }]

STEP PASSED

STEP 12 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 13 - Check if GetProfiles returned at least one MediaProfile item

STEP PASSED

STEP 14 - Check if MediaProfile does not contain configurations

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 15 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 16 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 17 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 18 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 19 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_1) }]

STEP PASSED

Restore profile 'ProfileToken_2' used for test

STEP 20 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 21 - Checking the DUT returned single MediaProfile

STEP PASSED

STEP 22 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1) }]

STEP PASSED

STEP 23 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 24 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoEncoder (VideoEncoderConfigurationToken_2) }]

STEP PASSED

TEST PASSED

MEDIA2-1-1-7-v20.06 FIXED MEDIA PROFILE CONFIGURATION

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { All }]

STEP PASSED

STEP 7 - Remove Configuration (Media2) [ProfileToken = ProfileToken_1, Configuration = { All }]

STEP PASSED

STEP 8 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 9 - Check if the DUT deleted all the configurations

STEP PASSED

STEP 10 - Add Configuration (Media2) [ProfileToken = ProfileToken_1, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1), VideoEncoder (VideoEncoderConfigurationToken_1), Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 11 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 12 - Check if the DUT returned profiles with the same configurations

STEP PASSED

STEP 13 - Remove Configuration (Media2) [ProfileToken = ProfileToken_2, Configuration = { All }]

STEP PASSED

STEP 14 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 15 - Check if the DUT deleted all the configurations

STEP PASSED

STEP 16 - Add Configuration (Media2) [ProfileToken = ProfileToken_2, no Name, Configuration = { VideoSource (VideoSourceConfigurationToken_1), VideoEncoder (VideoEncoderConfigurationToken_2), Metadata (MetadataConfigurationToken_1) }]

STEP PASSED

STEP 17 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 18 - Check if the DUT returned profiles with the same configurations

STEP PASSED

Restore profile 'ProfileToken_1' used for test

STEP 19 - Get Profiles (Media2) [Token = ProfileToken_1, Type = { All }]

STEP PASSED

STEP 20 - Checking the DUT returned single MediaProfile

STEP PASSED

Restore profile 'ProfileToken_2' used for test

STEP 21 - Get Profiles (Media2) [Token = ProfileToken_2, Type = { All }]

STEP PASSED

STEP 22 - Checking the DUT returned single MediaProfile

STEP PASSED

TEST PASSED

MEDIA2-2-2-1-v20.06 GET VIDEO SOURCE CONFIGURATION OPTIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 8 - The DUT returned no VideoSourceConfigurationOptions.BoundsRange items

STEP PASSED

STEP 9 - Check BoundsRange.HeightRange.Min <= BoundsRange.HeightRange.Max

STEP PASSED

STEP 10 - Check BoundsRange.WidthRange.Min <= BoundsRange.WidthRange.Max

STEP PASSED

STEP 11 - Check BoundsRange.XRange.Min <= BoundsRange.XRange.Max

STEP PASSED

STEP 12 - Check BoundsRange.YRange.Min <= BoundsRange.YRange.Max

STEP PASSED

STEP 13 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 14 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 15 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 16 - Get Profiles (Media2) [no Token, Type = { VideoSource }]

STEP PASSED

STEP 17 - Check the DUT returned at least one MediaProfile item

STEP PASSED

STEP 18 - Get Video Source Configuration Options (Media2)

STEP PASSED

TEST PASSED

MEDIA2-2-2-2-v17.01 GET VIDEO SOURCE CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 8 - Check all VideoSourceConfiguration items have unique tokens

STEP PASSED

STEP 9 - Get Video Source Configurations (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 10 - Check the DUT returned only single VideoSourceConfiguration item

STEP PASSED

STEP 11 - Check returned VideoSourceConfiguration item has the value of 'token' field as specified in 'GetVideoSourceConfigurations' request

STEP PASSED

STEP 12 - Get Profiles (Media2) [no Token, Type = { VideoSource }]

STEP PASSED

STEP 13 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_1]

STEP PASSED

STEP 14 - Check all VideoSourceConfiguration items have unique tokens

STEP PASSED

STEP 15 - Check complete VideoSourceConfiguration list has at least one item with the value of 'token' field as item from 'GetVideoSourceConfigurations' response

STEP PASSED

STEP 16 - Check MediaProfile.Configurations.VideoSource is present in complete list of VideoSourceConfiguration items

STEP PASSED

STEP 17 - Get Video Source Configurations (Media2) [no ConfigurationToken, ProfileToken = ProfileToken_2]

STEP PASSED

STEP 18 - Check all VideoSourceConfiguration items have unique tokens

STEP PASSED

STEP 19 - Check complete VideoSourceConfiguration list has at least one item with the value of 'token' field as item from 'GetVideoSourceConfigurations' response

STEP PASSED

STEP 20 - Check MediaProfile.Configurations.VideoSource is present in complete list of VideoSourceConfiguration items

STEP PASSED

TEST PASSED

MEDIA2-2-2-3-v17.01 VIDEO SOURCE CONFIGURATIONS AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 8 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 9 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 10 - Check VideoSourceConfiguration.SourceToken value is present in

VideoSourceConfigurationOptions.VideoSourceTokensAvailable list

STEP PASSED

STEP 11 - Check Options.BoundsRange.XRange.Min \leq VideoSourceConfiguration.Bounds.x

STEP PASSED

STEP 12 - Check VideoSourceConfiguration.Bounds.x \leq Options.BoundsRange.XRange.Max

STEP PASSED

STEP 13 - Check Options.BoundsRange.YRange.Min \leq VideoSourceConfiguration.Bounds.y

STEP PASSED

STEP 14 - Check VideoSourceConfiguration.Bounds.y \leq Options.BoundsRange.YRange.Max

STEP PASSED

STEP 15 - Check Options.BoundsRange.WidthRange.Min \leq VideoSourceConfiguration.Bounds.width

STEP PASSED

STEP 16 - Check VideoSourceConfiguration.Bounds.width \leq Options.BoundsRange.WidthRange.Max

STEP PASSED

STEP 17 - Check Options.BoundsRange.HeightRange.Min \leq VideoSourceConfiguration.Bounds.height

STEP PASSED

STEP 18 - Check VideoSourceConfiguration.Bounds.height \leq Options.BoundsRange.HeightRange.Max

STEP PASSED

TEST PASSED

MEDIA2-2-2-4-v17.01 PROFILES AND VIDEO SOURCE CONFIGURATIONS CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource }]

STEP PASSED

STEP 7 - Get Video Source Configurations (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 8 - Check the DUT returned the same VideoSourceConfiguration as was returned in 'GetProfiles' response

STEP PASSED

STEP 9 - Get Video Source Configurations (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 10 - Check the DUT returned the same VideoSourceConfiguration as was returned in 'GetProfiles' response

STEP PASSED

TEST PASSED

MEDIA2-2-2-5-v21.12 MODIFY ALL SUPPORTED VIDEO SOURCE CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 16 - Set Video Source Configuration (Media2)

STEP PASSED

STEP 17 - Send PullMessages request

STEP PASSED

STEP 18 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 19 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 20 - Response is not empty

STEP PASSED

STEP 21 - Waiting for notification

STEP PASSED

STEP 22 - Get Video Source Configurations (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 23 - Check the DUT returned only single VideoSourceConfiguration item

STEP PASSED

STEP 24 - Check returned VideoSourceConfiguration item has the value of 'token' field as specified in 'GetVideoSourceConfigurations' request

STEP PASSED

STEP 25 - Compare VideoSourceConfigurations before and after 'SetVideoSourceConfiguration' request

STEP PASSED

STEP 26 - Set Video Source Configuration (Media2)

STEP PASSED

STEP 27 - Send PullMessages request

STEP PASSED

STEP 28 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 29 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 30 - Response is not empty

STEP PASSED

STEP 31 - Waiting for notification

STEP PASSED

STEP 32 - Get Video Source Configurations (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 33 - Check the DUT returned only single VideoSourceConfiguration item

STEP PASSED

STEP 34 - Check returned VideoSourceConfiguration item has the value of 'token' field as specified in 'GetVideoSourceConfigurations' request

STEP PASSED

STEP 35 - Compare VideoSourceConfigurations before and after 'SetVideoSourceConfiguration' request

STEP PASSED

STEP 36 - Set Video Source Configuration (Media2)

STEP PASSED

STEP 37 - Send Unsubscribe request

STEP PASSED

STEP 38 - Get Video Source Configuration Options (Media2)

STEP PASSED

TEST PASSED

MEDIA2-2-2-6-v17.01 GET VIDEO SOURCE CONFIGURATIONS – INVALID TOKEN

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [ConfigurationToken = M, no ProfileToken]

STEP PASSED

TEST PASSED

MEDIA2-2-2-7-v17.12 PROFILES AND VIDEO SOURCE CONFIGURATION OPTIONS CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource }]

STEP PASSED

STEP 7 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 8 - Check if Video Source Configuration Options contains Video Source Token item is equal to Video Source Token item in the profile with token 'ProfileToken_1'

STEP PASSED

STEP 9 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds x item is greater than or equal to Bounds Range XRange Min item in Video Source Configuration Options

STEP PASSED

STEP 10 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds x item is less than or equal to Bounds Range XRange Max item in Video Source Configuration Options

STEP PASSED

STEP 11 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds y item is greater than or equal to Bounds Range YRange Min item in Video Source Configuration Options

STEP PASSED

STEP 12 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds y item is less than or equal to Bounds Range YRange Max item in Video Source Configuration Options

STEP PASSED

STEP 13 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds width item is greater than or equal to Bounds Range WidthRange Min item in Video Source Configuration Options

STEP PASSED

STEP 14 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds width item is less than or equal to Bounds Range WidthRange Max item in Video Source Configuration Options

STEP PASSED

STEP 15 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds height item is greater than or equal to Bounds Range HeightRange Min item in Video Source Configuration Options

STEP PASSED

STEP 16 - Check if the profile with token 'ProfileToken_1' contains Video Source Bounds height item is less than or equal to Bounds Range HeightRange Max item in Video Source Configuration Options

STEP PASSED

STEP 17 - Get Video Source Configuration Options (Media2)

STEP PASSED

STEP 18 - Check if Video Source Configuration Options contains Video Source Token item is equal to Video Source Token item in the profile with token 'ProfileToken_2'

STEP PASSED

STEP 19 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds x item is greater than or equal to Bounds Range XRange Min item in Video Source Configuration Options

STEP PASSED

STEP 20 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds x item is less than or equal to Bounds Range XRange Max item in Video Source Configuration Options

STEP PASSED

STEP 21 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds y item is greater than or equal to Bounds Range YRange Min item in Video Source Configuration Options

STEP PASSED

STEP 22 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds y item is less than or equal to Bounds Range YRange Max item in Video Source Configuration Options

STEP PASSED

STEP 23 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds width item is greater than or equal to Bounds Range WidthRange Min item in Video Source Configuration Options

STEP PASSED

STEP 24 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds width item is less than or equal to Bounds Range WidthRange Max item in Video Source Configuration Options

STEP PASSED

STEP 25 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds height item is greater than or equal to Bounds Range HeightRange Min item in Video Source Configuration Options

STEP PASSED

STEP 26 - Check if the profile with token 'ProfileToken_2' contains Video Source Bounds height item is less than or equal to Bounds Range HeightRange Max item in Video Source Configuration Options

STEP PASSED

TEST PASSED

MEDIA2-2-3-1-v20.12 VIDEO ENCODER CONFIGURATION

TestResult

STEP 1 - Getting media 2 service address

STEP PASSED

STEP 2 - Connect to Media 2 service

STEP PASSED

STEP 3 - Get Video Encoder Configurations

STEP PASSED

STEP 4 - Check if the DUT has video encoder configurations

STEP PASSED

STEP 5 - Check if the DUT has video encoder configurations with unique tokens

STEP PASSED

STEP 6 - GetProfiles

STEP PASSED

STEP 7 - Check GetProfilesResponse

STEP PASSED

STEP 8 - Get Video Encoder Configurations profile token=ProfileToken_1

STEP PASSED

STEP 9 - Check if the DUT has video encoder configurations with unique tokens

STEP PASSED

STEP 10 - Check if the current video encoder configuration from media profile listed in compatible video encoder configurations

STEP PASSED

STEP 11 - Check if all of compatible video encoder configurations are listed in total list of video encoder configurations

STEP PASSED

STEP 12 - Get Video Encoder Configurations profile token=ProfileToken_2

STEP PASSED

STEP 13 - Check if the DUT has video encoder configurations with unique tokens

STEP PASSED

STEP 14 - Check if the current video encoder configuration from media profile listed in compatible video encoder configurations

STEP PASSED

STEP 15 - Check if all of compatible video encoder configurations are listed in total list of video encoder configurations

STEP PASSED

STEP 16 - Get Video Encoder Configurations configuration token=VideoEncoderConfigurationToken_1

STEP PASSED

STEP 17 - Check if the DUT returned only one video encoder configuration with token 'VideoEncoderConfigurationToken_1'

STEP PASSED

STEP 18 - Get Video Encoder Configurations configuration token=VideoEncoderConfigurationToken_2

STEP PASSED

STEP 19 - Check if the DUT returned only one video encoder configuration with token 'VideoEncoderConfigurationToken_2'

STEP PASSED

TEST PASSED

MEDIA2-2-3-2-v20.12 VIDEO ENCODER CONFIGURATIONS AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY VALIDATION

TestResult

STEP 1 - Getting media 2 service address

STEP PASSED

STEP 2 - Connect to Media 2 service

STEP PASSED

STEP 3 - Get Video Encoder Configurations

STEP PASSED

STEP 4 - Get Video Encoder Configuration Options configuration token=VideoEncoderConfigurationToken_1

STEP PASSED

STEP 5 - Check if the DUT has consistent options for the configuration

STEP PASSED

STEP 6 - Get Video Encoder Configuration Options configuration token=VideoEncoderConfigurationToken_2

STEP PASSED

STEP 7 - Check if the DUT has consistent options for the configuration

STEP PASSED

TEST PASSED

MEDIA2-2-3-3-v20.12 PROFILES AND VIDEO ENCODER CONFIGURATION OPTIONS CONSISTENCY VALIDATION

TestResult

STEP 1 - Getting media 2 service address

STEP PASSED

STEP 2 - Connect to Media 2 service

STEP PASSED

STEP 3 - GetProfiles

STEP PASSED

STEP 4 - Check GetProfilesResponse

STEP PASSED

STEP 5 - Get Video Encoder Configuration Options profile token=ProfileToken_1 configuration
token=VideoEncoderConfigurationToken_1

STEP PASSED

STEP 6 - Check if the DUT has consistent options for the configuration

STEP PASSED

STEP 7 - Get Video Encoder Configuration Options profile token=ProfileToken_2 configuration
token=VideoEncoderConfigurationToken_2

STEP PASSED

STEP 8 - Check if the DUT has consistent options for the configuration

STEP PASSED

TEST PASSED

MEDIA2-2-3-4-v20.12 SET ALL SUPPORTED VIDEO ENCODER CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one Video Encoder configuration

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 16 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 17 - Send PullMessages request

STEP PASSED

STEP 18 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 19 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 20 - Response is not empty

STEP PASSED

STEP 21 - Waiting for notification

STEP PASSED

STEP 22 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 23 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 24 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 25 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 26 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 27 - Send PullMessages request

STEP PASSED

STEP 28 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 29 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 30 - Response is not empty

STEP PASSED

STEP 31 - Waiting for notification

STEP PASSED

STEP 32 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 33 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 34 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 35 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 36 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 37 - Send PullMessages request

STEP PASSED

STEP 38 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 39 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 40 - Response is not empty

STEP PASSED

STEP 41 - Waiting for notification

STEP PASSED

STEP 42 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 43 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 44 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 45 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 46 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 47 - Send PullMessages request

STEP PASSED

STEP 48 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 49 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 50 - Response is not empty

STEP PASSED

STEP 51 - Waiting for notification

STEP PASSED

STEP 52 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 53 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 54 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 55 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 56 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 57 - Send PullMessages request

STEP PASSED

STEP 58 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 59 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 60 - Response is not empty

STEP PASSED

STEP 61 - Waiting for notification

STEP PASSED

STEP 62 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 63 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 64 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 65 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 66 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 67 - Send PullMessages request

STEP PASSED

STEP 68 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 69 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 70 - Response is not empty

STEP PASSED

STEP 71 - Waiting for notification

STEP PASSED

STEP 72 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 73 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 74 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 75 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 76 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 77 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 78 - Send PullMessages request

STEP PASSED

STEP 79 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 80 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 81 - Response is not empty

STEP PASSED

STEP 82 - Waiting for notification

STEP PASSED

STEP 83 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 84 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 85 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 86 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 87 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 88 - Send PullMessages request

STEP PASSED

STEP 89 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 90 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 91 - Response is not empty

STEP PASSED

STEP 92 - Waiting for notification

STEP PASSED

STEP 93 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 94 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 95 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 96 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 97 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 98 - Send PullMessages request

STEP PASSED

STEP 99 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 100 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 101 - Response is not empty

STEP PASSED

STEP 102 - Waiting for notification

STEP PASSED

STEP 103 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 104 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 105 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 106 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 107 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 108 - Send PullMessages request

STEP PASSED

STEP 109 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 110 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 111 - Response is not empty

STEP PASSED

STEP 112 - Waiting for notification

STEP PASSED

STEP 113 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 114 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 115 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 116 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 117 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 118 - Send PullMessages request

STEP PASSED

STEP 119 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 120 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 121 - Response is not empty

STEP PASSED

STEP 122 - Waiting for notification

STEP PASSED

STEP 123 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 124 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 125 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 126 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 127 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 128 - Send PullMessages request

STEP PASSED

STEP 129 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 130 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 131 - Response is not empty

STEP PASSED

STEP 132 - Waiting for notification

STEP PASSED

STEP 133 - Get Video Encoder Configurations (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 134 - Check the DUT returned only single VideoEncoderConfiguration item

STEP PASSED

STEP 135 - Check returned VideoEncoderConfiguration item has the value of 'token' field as specified in 'GetVideoEncoderConfigurations' request

STEP PASSED

STEP 136 - Compare VideoEncoderConfigurations before and after 'SetVideoEncoderConfiguration' request

STEP PASSED

STEP 137 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1]

STEP PASSED

STEP 138 - Set Video Encoder Configuration (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2]

STEP PASSED

STEP 139 - Send Unsubscribe request

STEP PASSED

TEST PASSED

MEDIA2-2-3-5-v20.12 VIDEO ENCODER CONFIGURATION OPTIONS VALIDATION

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Encoder Configuration Options (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 8 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 9 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 10 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 11 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 12 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 13 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 14 - Get Video Encoder Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 15 - Check the DUT returned at least one Video Encoder configuration

STEP PASSED

STEP 16 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_1, no ProfileToken]

STEP PASSED

STEP 17 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 18 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 19 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 20 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 21 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 22 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 23 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 24 - Get Video Encoder Configuration Options (Media2) [ConfigurationToken = VideoEncoderConfigurationToken_2, no ProfileToken]

STEP PASSED

STEP 25 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 26 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 27 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 28 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

STEP 29 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with only two values

STEP PASSED

STEP 30 - Check the DUT returned Gov Length Range list in Video Encoder Configuration Options with the first value is less than the second

STEP PASSED

STEP 31 - Check the DUT returned Frame Rates Supported list in Video Encoder Configuration Options sorted with descending sort order

STEP PASSED

TEST PASSED

MEDIA2-2-4-1-v17.01 GET VIDEO SOURCE MODES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Device I/O service address

STEP PASSED

STEP 5 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 6 - Get Video Sources

STEP PASSED

STEP 7 - Checking the DUT returned at least one VideoSource item

STEP PASSED

STEP 8 - Get Media2 service address

STEP PASSED

STEP 9 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 10 - Get Video Source Modes (Media2)

STEP PASSED

STEP 11 - Check the DUT returned at least one VideoSourceMode item

STEP PASSED

STEP 12 - Check the DUT did not return VideoSourceMode items with the same tokens

STEP PASSED

STEP 13 - Check the DUT returned only one VideoSourceMode item with Enabled is equal to true

STEP PASSED

STEP 14 - Check the DUT did not return VideoSourceMode items with empty Encodings list

STEP PASSED

TEST PASSED

MEDIA2-2-4-2-v21.06 SET VIDEO SOURCE MODES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Device I/O service address

STEP PASSED

STEP 5 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 6 - Get Video Sources

STEP PASSED

STEP 7 - Checking the DUT returned at least one VideoSource item

STEP PASSED

STEP 8 - Get Media2 service address

STEP PASSED

STEP 9 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 10 - Get Video Source Modes (Media2)

STEP PASSED

STEP 11 - Check the DUT returned at least one VideoSourceMode item

STEP PASSED

STEP 12 - Set Video Source Mode (Media2)

STEP PASSED

STEP 13 - Get Video Source Modes (Media2)

STEP PASSED

STEP 14 - Check the DUT returned at least one VideoSourceMode item

STEP PASSED

STEP 15 - Check the DUT returned only one VideoSourceMode item with Enabled is equal to true

STEP PASSED

STEP 16 - Check the DUT returned VideoSourceMode item with the same token as in SetVideoSourceMode

STEP PASSED

STEP 17 - Check VideoSourceMode item has Enabled is equal to true

STEP PASSED

TEST PASSED

MEDIA2-5-1-1-v20.12 SNAPSHOT URI

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { VideoSource, VideoEncoder }]

STEP PASSED

STEP 7 - Checking the DUT returned at least one MediaProfile item

STEP PASSED

STEP 8 - Get Snapshot Uri (Media2)

STEP PASSED

STEP 9 - Check GetSnapshotUriResponse message is returned

STEP PASSED

STEP 10 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/snapshot/ProfileToken_1'

STEP PASSED

STEP 11 - Check HTTP status code

STEP PASSED

STEP 12 - Check JPEG image data is returned

STEP PASSED

TEST PASSED

MEDIA2-5-1-2-v20.12 VIDEO ENCODER INSTANCES PER VIDEO SOURCE

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration

STEP PASSED

STEP 8 - Get Device I/O service address

STEP PASSED

STEP 9 - Check that the DUT returned Device I/O service address

STEP PASSED

STEP 10 - Get Video Sources

STEP PASSED

STEP 11 - Get Video Encoder Instances (Media2) [ConfigurationToken = VideoSourceConfigurationToken_1]

STEP PASSED

STEP 12 - Check the DUT returned at least one VideoSourceConfiguration with 'SourceToken' = 'VideoSourceToken_1' for which the GetVideoEncoderInstances returns a Total greater than 0

STEP PASSED

TEST PASSED

MEDIA2-6-1-1-v18.06 CREATE OSD CONFIGURATION FOR TEXT OVERLAY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration

STEP PASSED

STEP 8 - Get OSDs (Media2)

STEP PASSED

STEP 9 - Get OSD Options (Media2)

STEP PASSED

STEP 10 - Delete OSD (Media2)

STEP PASSED

STEP 11 - Check the DUT returned OSDConfigurationOptions with TextOption

STEP PASSED

STEP 12 - Create OSD (Media2)

STEP PASSED

STEP 13 - Check the DUT just created OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 14 - Get OSDs (Media2)

STEP PASSED

STEP 15 - Check the DUT returned the OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 16 - Check the DUT returned the same OSDConfiguration as was sent in 'CreateOSD' request

STEP PASSED

STEP 17 - Delete OSD (Media2)

STEP PASSED

STEP 18 - Check the DUT returned OSDConfigurationOptions with TextOption

STEP PASSED

STEP 19 - Check the DUT returned OSDConfigurationOptions with DateFormat

STEP PASSED

STEP 20 - Check the DUT returned OSDConfigurationOptions with TimeFormat

STEP PASSED

STEP 21 - Create OSD (Media2)

STEP PASSED

STEP 22 - Check the DUT just created OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 23 - Get OSDs (Media2)

STEP PASSED

STEP 24 - Check the DUT returned the OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 25 - Check the DUT returned the same OSDConfiguration as was sent in 'CreateOSD' request

STEP PASSED

STEP 26 - Delete OSD (Media2)

STEP PASSED

STEP 27 - Get OSDs (Media2)

STEP PASSED

STEP 28 - Check the DUT removed just created OSDConfiguration

STEP PASSED

STEP 29 - Create OSD (Media2)

STEP PASSED

TEST PASSED

MEDIA2-6-1-2-v20.06 CREATE OSD CONFIGURATION FOR IMAGE OVERLAY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration

STEP PASSED

STEP 8 - Get OSDs (Media2)

STEP PASSED

STEP 9 - Get OSD Options (Media2)

STEP PASSED

TEST PASSED

MEDIA2-6-1-3-v20.06 SET OSD CONFIGURATION IMAGE OVERLAY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration

STEP PASSED

STEP 8 - Get OSDs (Media2)

STEP PASSED

STEP 9 - Get OSD Options (Media2)

STEP PASSED

TEST PASSED

MEDIA2-6-1-4-v18.06 SET OSD CONFIGURATION TEXT OVERLAY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration

STEP PASSED

STEP 8 - Get OSDs (Media2)

STEP PASSED

STEP 9 - Get OSD Options (Media2)

STEP PASSED

STEP 10 - Delete OSD (Media2)

STEP PASSED

STEP 11 - Check the DUT returned OSDConfigurationOptions with TextOption

STEP PASSED

STEP 12 - Create OSD (Media2)

STEP PASSED

STEP 13 - Get OSDs (Media2)

STEP PASSED

STEP 14 - Check the DUT returned the OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 15 - Check the DUT returned OSDConfigurationOptions with TextOption

STEP PASSED

STEP 16 - Set OSD (Media2)

STEP PASSED

STEP 17 - Get OSDs (Media2)

STEP PASSED

STEP 18 - Check the DUT returned the OSDConfiguration with token from 'CreateOSD' response

STEP PASSED

STEP 19 - Check the DUT returned the same OSDConfiguration as was sent in 'SetOSD' request

STEP PASSED

STEP 20 - Delete OSD (Media2)

STEP PASSED

STEP 21 - Create OSD (Media2)

STEP PASSED

TEST PASSED

MEDIA2-6-1-5-v17.12 GET OSDS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get OSDs (Media2)

STEP PASSED

STEP 7 - Check if OSDConfigurations list does not contain items with the same token

STEP PASSED

STEP 8 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 9 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 10 - Get OSDs (Media2)

STEP PASSED

STEP 11 - Check if OSDConfigurations list does not contain items with the same token

STEP PASSED

STEP 12 - Check if OSDConfigurations list from the first GetOSDs call contains the entire OSDConfigurations list from GetOSDs call with 'VideoSourceConfigurationToken_1' parameter token

STEP PASSED

TEST PASSED

MEDIA2-6-1-6-v18.06 GET OSD OPTIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 8 - Get OSD Options (Media2)

STEP PASSED

STEP 9 - Check if OSDConfigurationOptions item contains Type field with value 'Text' and non-empty TextOption field or does not contain these fields

STEP PASSED

STEP 10 - Check if OSDConfigurationOptions item contains Type field with value 'Image' and non-empty ImageOption field or does not contain these fields

STEP PASSED

STEP 11 - Check if OSDConfigurationOptions item contains Type field with value 'Text' when

MaximumNumberOfOSDs.@PlainText > 0

STEP PASSED

STEP 12 - Check if OSDConfigurationOptions item contains TextOption.Type field with value 'Date' and TextOption.DateFormat field

STEP PASSED

STEP 13 - Check if OSDConfigurationOptions item contains TextOption.Type field with value 'Time' and

TextOption.TimeFormat field

STEP PASSED

STEP 14 - Check if OSDConfigurationOptions item contains TextOption.Type field with value 'DateAndTime' and

TextOption.TimeFormat and TextOption.DateFormat fields

STEP PASSED

STEP 15 - Check if at least one OSDConfigurationOptions item contains MaximumNumberOfOSDs.Total > 0

STEP PASSED

TEST PASSED

MEDIA2-6-1-7-v18.06 OSD CONFIGURATIONS AND OSD OPTIONS CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Video Source Configurations (Media2) [no ConfigurationToken, no ProfileToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one VideoSourceConfiguration item

STEP PASSED

STEP 8 - Get OSDs (Media2)

STEP PASSED

STEP 9 - Get OSD Options (Media2)

STEP PASSED

STEP 10 - Check that VideoSourceConfigurationToken field in OSDConfiguration item is equal to token field in VideoSourceConfiguration item

STEP PASSED

STEP 11 - Check if OSDConfigurationOptions item contains MaximumNumberOfOSDs.Total > 0

STEP PASSED

STEP 12 - Check that in OSDConfigurationOptions item Type field contains the value is equal to Type value in OSDConfiguration item

STEP PASSED

STEP 13 - Check that in OSDConfigurationOptions item PositionOption field contains the value is equal to Position Type field in OSDConfiguration item

STEP PASSED

STEP 14 - Check that in OSDConfigurationOptions item TextOption Type field contains the value is equal to TextString Type field in OSDConfiguration item

STEP PASSED

TEST PASSED

MEDIA2-7-1-1-v18.12 MEDIA2 SERVICE CAPABILITIES

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Service Capabilities(Media2)

STEP PASSED

STEP 7 - Check the DUT returned MaximumNumberOfProfiles in ProfileCapabilities in the service capabilities response

STEP PASSED

STEP 8 - Check the DUT returned ConfigurationsSupported in ProfileCapabilities in the service capabilities response

STEP PASSED

STEP 9 - Check the DUT returned at least one item in ConfigurationsSupported in the service capabilities response

STEP PASSED

STEP 10 - Check the DUT does not return 'All' item in ConfigurationsSupported in the service capabilities response

STEP PASSED

STEP 11 - Check if service capabilities response contains 'VideoSource' in ConfigurationsSupported list

STEP PASSED

TEST PASSED

MEDIA2-7-1-2-v17.06 GET SERVICES AND GET MEDIA2 SERVICE CAPABILITIES CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Check Media2 service is supported

STEP PASSED

STEP 5 - Check service capabilities is present for Media2 service

STEP PASSED

STEP 6 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 7 - Check that Media2 Capabilities are found

STEP PASSED

STEP 8 - Get Media2 service address

STEP PASSED

STEP 9 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 10 - Get Service Capabilities(Media2)

STEP PASSED

STEP 11 - Check Media2ServiceCapabilities consistency

STEP PASSED

TEST PASSED

MEDIA2-8-1-1-v20.12 MODIFY ALL SUPPORTED METADATA CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Metadata Configurations (Media2) [no ProfileToken, no ConfigurationToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one MetadataConfiguration item

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Get Metadata Configuration Options (Media2) [no ProfileToken, ConfigurationToken =

MetadataConfigurationToken_1]

STEP PASSED

STEP 16 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 17 - Send PullMessages request

STEP PASSED

STEP 18 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 19 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 20 - Response is not empty

STEP PASSED

STEP 21 - Waiting for tns1:Media/ConfigurationChanged notification

STEP PASSED

STEP 22 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 23 - Check the DUT returned only single MetadataConfiguration item

STEP PASSED

STEP 24 - Check returned MetadataConfiguration item has the value of 'token' field as specified in 'GetMetadataConfigurations' request

STEP PASSED

STEP 25 - Check MetadataConfiguration after 'SetMetadataConfiguration' request [token = MetadataConfigurationToken_1]

STEP PASSED

STEP 26 - Set Metadata Configuration (Media2)

STEP PASSED

STEP 27 - Send PullMessages request

STEP PASSED

STEP 28 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 29 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 30 - Response is not empty

STEP PASSED

STEP 31 - Waiting for tns1:Media/ConfigurationChanged notification

STEP PASSED

STEP 32 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 33 - Check the DUT returned only single MetadataConfiguration item

STEP PASSED

STEP 34 - Check returned MetadataConfiguration item has the value of 'token' field as specified in 'GetMetadataConfigurations' request

STEP PASSED

STEP 35 - Check MetadataConfiguration after 'SetMetadataConfiguration' request [token = MetadataConfigurationToken_1]

STEP PASSED

STEP 36 - Send Unsubscribe request

STEP PASSED

STEP 37 - Set Metadata Configuration (Media2)

STEP PASSED

TEST PASSED

MEDIA2-8-1-2-v19.12 GET METADATA CONFIGURATIONS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Metadata Configurations (Media2) [no ProfileToken, no ConfigurationToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one MetadataConfiguration item

STEP PASSED

STEP 8 - Check all MetadataConfiguration items have unique tokens

STEP PASSED

STEP 9 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 10 - Check the DUT returned only single MetadataConfiguration item

STEP PASSED

STEP 11 - Check returned MetadataConfiguration item has the value of 'token' field as specified in 'GetMetadataConfigurations' request

STEP PASSED

STEP 12 - Get Profiles (Media2) [no Token, Type = { Metadata }]

STEP PASSED

STEP 13 - Get Metadata Configurations (Media2) [ProfileToken = ProfileToken_1, no ConfigurationToken]

STEP PASSED

STEP 14 - Check all MetadataConfiguration items have unique tokens

STEP PASSED

STEP 15 - Check complete MetadataConfiguration list has at least one item with the value of 'token' field as item from 'GetMetadataConfigurations' response

STEP PASSED

STEP 16 - Check MediaProfile.Configurations.Metadata is present in compatible list of MetadataConfiguration items

STEP PASSED

STEP 17 - Get Metadata Configurations (Media2) [ProfileToken = ProfileToken_2, no ConfigurationToken]

STEP PASSED

STEP 18 - Check all MetadataConfiguration items have unique tokens

STEP PASSED

STEP 19 - Check complete MetadataConfiguration list has at least one item with the value of 'token' field as item from 'GetMetadataConfigurations' response

STEP PASSED

STEP 20 - Check MediaProfile.Configurations.Metadata is present in compatible list of MetadataConfiguration items

STEP PASSED

TEST PASSED

MEDIA2-8-1-3-v19.12 PROFILES AND METADATA CONFIGURATIONS CONSISTENCY

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Profiles (Media2) [no Token, Type = { Metadata }]

STEP PASSED

STEP 7 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 8 - Check the DUT returned the same MetadataConfiguration as was returned in 'GetProfiles' response

STEP PASSED

STEP 9 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = MetadataConfigurationToken_1]

STEP PASSED

STEP 10 - Check the DUT returned the same MetadataConfiguration as was returned in 'GetProfiles' response

STEP PASSED

TEST PASSED

MEDIA2-8-1-4-v19.12 GET METADATA CONFIGURATIONS – INVALID TOKEN

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Media2 service address

STEP PASSED

STEP 5 - Check that the DUT returned Media2 service address

STEP PASSED

STEP 6 - Get Metadata Configurations (Media2) [no ProfileToken, no ConfigurationToken]

STEP PASSED

STEP 7 - Check the DUT returned at least one MetadataConfiguration item

STEP PASSED

STEP 8 - Get Metadata Configurations (Media2) [no ProfileToken, ConfigurationToken = x]

STEP PASSED

TEST PASSED

Security Test Cases

SECURITY-1-1-1-v14.12 USER TOKEN PROFILE

TestResult

STEP 1 - Check if credentials were defined

STEP PASSED

STEP 2 - Sending request to the DUT with omitted Nonce

STEP PASSED

STEP 3 - Sending request to the DUT with omitted Created

STEP PASSED

STEP 4 - Sending request to the DUT with omitted Password/Type

STEP PASSED

STEP 5 - Sending valid request to the DUT

STEP PASSED

TEST PASSED

SECURITY-1-1-2-v14.12 DIGEST AUTHENTICATION

TestResult

STEP 1 - Check if credentials were defined

STEP PASSED

STEP 2 - Invoke GetDeviceInformation without credentials supplied

STEP PASSED

STEP 3 - Check response

STEP PASSED

STEP 4 - Sending valid request to the DUT

STEP PASSED

TEST PASSED

IP Configuration

IPCONFIG-1-1-3-v21.06 IPV4 DHCP

TestResult

STEP 1 - Get network interfaces

STEP PASSED

STEP 2 - Check that the DUT returned current interfaces

STEP PASSED

STEP 3 - Verifying IPv4 presence

STEP PASSED

STEP 4 - Waiting for Hello message...

STEP PASSED

STEP 5 - Set network interface

STEP PASSED

STEP 6 - Send System Reboot message

STEP PASSED

STEP 7 - Waiting for Hello message from the DUT

STEP PASSED

STEP 8 - 5 seconds timeout after Hello

STEP PASSED

STEP 9 - Waiting for Hello message...

STEP PASSED

STEP 10 - Waiting for Hello message from the DUT

STEP 11 - Verifying Hello message

STEP PASSED

STEP 12 - Identifying right address

STEP PASSED

STEP 13 - Get network interfaces

STEP PASSED

STEP 14 - Verifying appliance of IPv4 static settings

STEP PASSED

STEP 15 - Waiting for Hello message...

STEP PASSED

STEP 16 - Restore network settings

STEP PASSED

STEP 17 - Send System Reboot message

STEP PASSED

STEP 18 - Waiting for Hello message from the DUT

STEP PASSED

STEP 19 - 5 seconds timeout after Hello

STEP PASSED

STEP 20 - Waiting for Hello message...

STEP PASSED

STEP 21 - Waiting for Hello message from the DUT

STEP 22 - Verifying Hello message

STEP PASSED

STEP 23 - Identifying right address

STEP PASSED

TEST PASSED

IPCONFIG-1-1-5-v20.12 IPV4 LINK LOCAL ADDRESS

TestResult

STEP 1 - Get network interfaces

STEP PASSED

STEP 2 - Check that the DUT returned current interfaces

STEP PASSED

STEP 3 - Verifying IPv4 presence

STEP PASSED

STEP 4 - Get Network Zero configuration

STEP PASSED

STEP 5 - Set Network Zero configuration

STEP PASSED

STEP 6 - 10 seconds timeout after SetZeroConfiguration

STEP PASSED

STEP 7 - Get Network Zero configuration

STEP PASSED

STEP 8 - Verifying appliance of IPv4 zero settings

STEP PASSED

STEP 9 - Set Network Zero configuration

STEP PASSED

TEST PASSED

Device Discovery

DISCOVERY-1-1-2-v21.06 HELLO MESSAGE VALIDATION

TestResult

STEP 1 - Send System Reboot message

STEP PASSED

STEP 2 - Waiting for Hello message...

STEP PASSED

STEP 3 - Waiting for Hello message from the DUT

STEP PASSED

STEP 4 - 5 seconds timeout after Hello

STEP PASSED

STEP 5 - Validating hello message

STEP PASSED

TEST PASSED

DISCOVERY-1-1-3-v21.06 SEARCH BASED ON DEVICE SCOPE TYPES

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-1-1-4-v21.06 SEARCH WITH OMITTED DEVICE AND SCOPE TYPES

TestResult

STEP 1 - Probe device

STEP PASSED

STEP 2 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-1-1-5-v21.06 RESPONSE TO INVALID SEARCH REQUEST

TestResult

STEP 1 - Probe device - negative test

STEP PASSED

TEST PASSED

DISCOVERY-1-1-6-v21.06 SEARCH USING UNICAST PROBE MESSAGE

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

STEP 5 - Probe device

STEP PASSED

STEP 6 - Validate probe match

STEP PASSED

STEP 7 - Probe device - negative test

STEP PASSED

TEST PASSED

DISCOVERY-1-1-8-v14.12 BYE MESSAGE

TestResult

STEP 1 - Waiting for Bye message...

STEP PASSED

STEP 2 - Reboot device

STEP PASSED

STEP 3 - Waiting for Bye message from the DUT

STEP PASSED

STEP 4 - Waiting for device to reboot

STEP PASSED

TEST PASSED

DISCOVERY-1-1-9-v21.06 DISCOVERY MODE CONFIGURATION

TestResult

STEP 1 - Get Discovery Mode

STEP PASSED

STEP 2 - Check current DiscoveryMode

STEP PASSED

STEP 3 - Set Discovery Mode

STEP PASSED

STEP 4 - Get Discovery Mode

STEP PASSED

STEP 5 - Check current DiscoveryMode

STEP PASSED

STEP 6 - Probe device - negative test

STEP PASSED

STEP 7 - Waiting for Bye or Hello message...

STEP PASSED

STEP 8 - Reboot device

STEP PASSED

STEP 9 - Waiting for Bye or Hello message from the DUT

STEP PASSED

STEP 10 - Set Discovery Mode

STEP PASSED

TEST PASSED

DISCOVERY-1-1-11-v21.06 DEVICE SCOPES CONFIGURATION

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Set device scopes

STEP PASSED

STEP 3 - Waiting for Hello message...

STEP PASSED

STEP 4 - Add device scopes

STEP PASSED

STEP 5 - Waiting for Hello message from the DUT

STEP PASSED

STEP 6 - 5 seconds timeout after Hello

STEP PASSED

STEP 7 - Hello message validation

STEP PASSED

STEP 8 - Probe device

STEP PASSED

STEP 9 - Validate probe match

STEP PASSED

STEP 10 - Waiting for Hello message...

STEP PASSED

STEP 11 - Remove device scopes

STEP PASSED

STEP 12 - Waiting for Hello message from the DUT

STEP PASSED

STEP 13 - 5 seconds timeout after Hello

STEP PASSED

STEP 14 - Hello message validation

STEP PASSED

STEP 15 - Probe device - negative test

STEP PASSED

STEP 16 - Set device scopes

STEP PASSED

TEST PASSED

DISCOVERY-2-1-1-v21.06 DISCOVERY - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-2-1-2-v21.06 DISCOVERY - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-2-1-3-v21.06 DISCOVERY - NAMESPACES (NOT STANDARD PREFIXES)

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-2-1-4-v21.06 DISCOVERY - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

DISCOVERY-2-1-5-v21.06 DISCOVERY - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)

TestResult

STEP 1 - Get device scopes

STEP PASSED

STEP 2 - Validating device scopes

STEP PASSED

STEP 3 - Probe device

STEP PASSED

STEP 4 - Validate probe match

STEP PASSED

TEST PASSED

Device Management

DEVICE-1-1-2-v14.12 ALL CAPABILITIES

TestResult

STEP 1 - Get capabilities

STEP PASSED

STEP 2 - Check capabilities

STEP PASSED

STEP 3 - Check that DUT returned Device capabilities

STEP PASSED

STEP 4 - Check that DUT returned Events capabilities

STEP PASSED

STEP 5 - Check that DUT returned DeviceIO capabilities

STEP PASSED

STEP 6 - Check that DUT returned Imaging capabilities

STEP PASSED

STEP 7 - Get capabilities

STEP PASSED

STEP 8 - Check capabilities

STEP PASSED

STEP 9 - Check that DUT returned Device capabilities

STEP PASSED

STEP 10 - Check that DUT returned Events capabilities

STEP PASSED

STEP 11 - Check that DUT returned DeviceIO capabilities

STEP PASSED

STEP 12 - Check that DUT returned Imaging capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-3-v14.12 DEVICE CAPABILITIES

TestResult

STEP 1 - Get capabilities

STEP PASSED

STEP 2 - Check that DUT returned capabilities

STEP PASSED

STEP 3 - Check that DUT returned device capabilities

STEP PASSED

STEP 4 - Validate device address (http://192.168.3.36:8000/onvif/device_service)

STEP PASSED

STEP 5 - Check that DUT returned network capabilities

STEP PASSED

STEP 6 - Check that DUT returned system capabilities

STEP PASSED

STEP 7 - Check that DUT did not return analytics capabilities

STEP PASSED

STEP 8 - Check that DUT did not return events capabilities

STEP PASSED

STEP 9 - Check that DUT did not return imaging capabilities

STEP PASSED

STEP 10 - Check that DUT did not return media capabilities

STEP PASSED

STEP 11 - Check that DUT did not return PTZ capabilities

STEP PASSED

STEP 12 - Check supported ONVIF versions

STEP PASSED

STEP 13 - Check that DUT returned IO capabilities

STEP PASSED

STEP 14 - Check that DUT returned security capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-4-v14.12 MEDIA CAPABILITIES

TestResult

STEP 1 - Get Media Capabilities - negative test

STEP PASSED

TEST PASSED

DEVICE-1-1-5-v14.12 EVENT CAPABILITIES

TestResult

STEP 1 - Get capabilities

STEP PASSED

STEP 2 - Check that DUT returned capabilities

STEP PASSED

STEP 3 - Check that DUT returned events capabilities

STEP PASSED

STEP 4 - Validate events address (http://192.168.3.36:8000/onvif/event_service)

STEP PASSED

STEP 5 - Check that DUT did not return device capabilities

STEP PASSED

STEP 6 - Check that DUT did not return analytics capabilities

STEP PASSED

STEP 7 - Check that DUT did not return imaging capabilities

STEP PASSED

STEP 8 - Check that DUT did not return media capabilities

STEP PASSED

STEP 9 - Check that DUT did not return PTZ capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-6-v14.12 PTZ CAPABILITIES

TestResult

STEP 1 - Get PTZ Capabilities - negative test

STEP PASSED

TEST PASSED

DEVICE-1-1-9-v14.12 SOAP FAULT MESSAGE

TestResult

STEP 1 - Get capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-10-v14.12 IMAGING CAPABILITIES

TestResult

STEP 1 - Get capabilities

STEP PASSED

STEP 2 - Check that DUT returned capabilities

STEP PASSED

STEP 3 - Check that DUT returned Imaging capabilities

STEP PASSED

STEP 4 - Validate imaging address (http://192.168.3.36:8000/onvif/image_service)

STEP PASSED

STEP 5 - Check that DUT did not return device capabilities

STEP PASSED

STEP 6 - Check that DUT did not return analytics capabilities

STEP PASSED

STEP 7 - Check that DUT did not return events capabilities

STEP PASSED

STEP 8 - Check that DUT did not return media capabilities

STEP PASSED

STEP 9 - Check that DUT did not return PTZ capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-11-v14.12 ANALYTICS CAPABILITIES

TestResult

STEP 1 - Get Analytics Capabilities - negative test

STEP PASSED

TEST PASSED

DEVICE-1-1-13-v14.12 GET SERVICES – DEVICE SERVICE

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that DUT returned Device service address

STEP PASSED

STEP 3 - Check that no Capabilities returned

STEP PASSED

STEP 4 - Get Services

STEP PASSED

STEP 5 - Check that DUT returned Device service address

STEP PASSED

STEP 6 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 7 - Check that Capabilities element is correct

STEP PASSED

TEST PASSED

DEVICE-1-1-16-v14.12 GET SERVICES – EVENT SERVICE

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that DUT returned Event service address

STEP PASSED

STEP 3 - Check that no Capabilities returned

STEP PASSED

STEP 4 - Get Services

STEP PASSED

STEP 5 - Check that DUT returned Event service address

STEP PASSED

STEP 6 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 7 - Check that Capabilities element is correct

STEP PASSED

TEST PASSED

DEVICE-1-1-17-v14.12 GET SERVICES – IMAGING SERVICE

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that DUT returned Imaging service address

STEP PASSED

STEP 3 - Check that no Capabilities returned

STEP PASSED

STEP 4 - Get Services

STEP PASSED

STEP 5 - Check that DUT returned Imaging service address

STEP PASSED

STEP 6 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 7 - Check that Capabilities element is correct

STEP PASSED

TEST PASSED

DEVICE-1-1-18-v21.06 DEVICE SERVICE CAPABILITIES

TestResult

STEP 1 - Get service capabilities

STEP PASSED

STEP 2 - Check if DeviceServiceCapabilities item contains System.DiscoveryNotSupported = false or System.DiscoveryBye = false

STEP PASSED

TEST PASSED

DEVICE-1-1-19-v21.06 GET SERVICES AND GET DEVICE SERVICE CAPABILITIES CONSISTENCY

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that the DUT returned Device service information

STEP PASSED

STEP 3 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 4 - Get service capabilities

STEP PASSED

STEP 5 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 6 - Compare Capabilities

STEP PASSED

TEST PASSED

DEVICE-1-1-30-v17.06 GET SERVICES AND GET CAPABILITIES CONSISTENCY

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Get capabilities

STEP PASSED

STEP 3 - Check that the DUT returned Device Management service information

STEP PASSED

STEP 4 - Check that the DUT returned Device Management service information

STEP PASSED

STEP 5 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 6 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 7 - Check that Network -> IPFilter capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 8 - Check that Network -> ZeroConfiguration capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 9 - Check that Network -> IPVersion6 capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 10 - Check that Network -> DynDNS capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 11 - Check that Network -> Dot11Configuration capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 12 - Check that System -> DiscoveryResolve capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 13 - Check that System -> DiscoveryBye capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 14 - Check that System -> DiscoveryBye capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 15 - Check that System -> RemoteDiscovery capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 16 - Check that System -> SystemBackup capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 17 - Check that System -> SystemLogging capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 18 - Check that System -> FirmwareUpgrade capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 19 - Check that System -> HttpFirmwareUpgrade capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 20 - Check that System -> HttpSystemBackup capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 21 - Check that System -> HttpSystemLogging capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 22 - Check that System -> HttpSupportInformation capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 23 - Check that Security -> TLS1.1 capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 24 - Check that Security -> TLS1.2 capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 25 - Check that Security -> OnboardKeyGeneration capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 26 - Check that Security -> AccessPolicyConfig capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 27 - Check that Security -> X.509Token capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 28 - Check that Security -> SAMLToken capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 29 - Check that Security -> KerberosToken capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 30 - Check that Security -> RELToken capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 31 - Check that Security -> TLS1.0 capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 32 - Check that Security -> Dot1X capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 33 - Check that Security -> SupportedEAPMethod capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 34 - Check that Security -> RemoteUserHandling capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 35 - Check that the DUT returned Events service information
STEP PASSED

STEP 36 - Check that the DUT returned Events service information
STEP PASSED

STEP 37 - Check that the DUT returned Capabilities element
STEP PASSED

STEP 38 - Parse Capabilities element in GetServices response
STEP PASSED

STEP 39 - Check that WSSubscriptionPolicySupport capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 40 - Check that WSPullPointSupport capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 41 - Check that WSPausableSubscriptionManagerInterfaceSupport capability has equal values in GetServices and in GetCapabilities response
STEP PASSED

STEP 42 - Check that the DUT returned Imaging service information
STEP PASSED

STEP 43 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 44 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 45 - Check that the DUT returned DeviceIO service information

STEP PASSED

STEP 46 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 47 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 48 - Check that VideoSources capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 49 - Check that VideoOutputs capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 50 - Check that AudioSources capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 51 - Check that AudioOutputs capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

STEP 52 - Check that RelayOutputs capability has equal values in GetServices and in GetCapabilities response

STEP PASSED

TEST PASSED

DEVICE-1-1-31-v18.12 GET SERVICES - XADDR

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Check if the service with namespace "http://www.onvif.org/ver10/device/wsdl" contains http address scheme

STEP PASSED

STEP 5 - Check if the service with namespace "http://www.onvif.org/ver10/device/wsdl" contains authority component

STEP PASSED

STEP 6 - Check if the service with namespace "http://www.onvif.org/ver10/device/wsdl" contains address 192.168.3.36

STEP PASSED

STEP 7 - Check if the service with namespace "http://www.onvif.org/ver10/device/wsdl" contains address with port 8000

STEP PASSED

STEP 8 - Check if the service with namespace "http://www.onvif.org/ver10/events/wsdl" contains http address scheme

STEP PASSED

STEP 9 - Check if the service with namespace "http://www.onvif.org/ver10/events/wsdl" contains authority component

STEP PASSED

STEP 10 - Check if the service with namespace "http://www.onvif.org/ver10/events/wsdl" contains address 192.168.3.36

STEP PASSED

STEP 11 - Check if the service with namespace "http://www.onvif.org/ver10/events/wsdl" contains address with port 8000

STEP PASSED

STEP 12 - Check if the service with namespace "http://www.onvif.org/ver20/media/wsdl" contains http address scheme

STEP PASSED

STEP 13 - Check if the service with namespace "http://www.onvif.org/ver20/media/wsdl" contains authority component

STEP PASSED

STEP 14 - Check if the service with namespace "http://www.onvif.org/ver20/media/wsdl" contains address 192.168.3.36

STEP PASSED

STEP 15 - Check if the service with namespace "http://www.onvif.org/ver20/media/wsdl" contains address with port 8000

STEP PASSED

STEP 16 - Check if the service with namespace "http://www.onvif.org/ver20/imaging/wsdl" contains http address scheme

STEP PASSED

STEP 17 - Check if the service with namespace "http://www.onvif.org/ver20/imaging/wsdl" contains authority component
STEP PASSED

STEP 18 - Check if the service with namespace "http://www.onvif.org/ver20/imaging/wsdl" contains address 192.168.3.36
STEP PASSED

STEP 19 - Check if the service with namespace "http://www.onvif.org/ver20/imaging/wsdl" contains address with port 8000
STEP PASSED

STEP 20 - Check if the service with namespace "http://www.onvif.org/ver10/deviceIO/wsdl" contains http address scheme
STEP PASSED

STEP 21 - Check if the service with namespace "http://www.onvif.org/ver10/deviceIO/wsdl" contains authority component
STEP PASSED

STEP 22 - Check if the service with namespace "http://www.onvif.org/ver10/deviceIO/wsdl" contains address 192.168.3.36
STEP PASSED

STEP 23 - Check if the service with namespace "http://www.onvif.org/ver10/deviceIO/wsdl" contains address with port 8000
STEP PASSED

TEST PASSED

DEVICE-2-1-1-v20.12 NETWORK COMMAND HOSTNAME CONFIGURATION

TestResult

STEP 1 - Get Hostname
STEP PASSED

STEP 2 - Check that hostname information returned from the DUT
STEP PASSED

STEP 3 - Validate hostname ('DESKTOP-CK2D8T0')
STEP PASSED

TEST PASSED

DEVICE-2-1-3-v20.12 NETWORK COMMAND SETHOSTNAME TEST ERROR CASE

TestResult

STEP 1 - Get Hostname

STEP PASSED

STEP 2 - Check that the DUT returned current hostname information

STEP PASSED

STEP 3 - Set Hostname - negative test

STEP PASSED

STEP 4 - Get Hostname

STEP PASSED

STEP 5 - Check that current hostname returned from the DUT

STEP PASSED

STEP 6 - Verify that hostname has not been changed

STEP PASSED

STEP 7 - Verify that FromDHCP has not been changed

STEP PASSED

TEST PASSED

DEVICE-2-1-4-v20.12 GET DNS CONFIGURATION

TestResult

STEP 1 - Get DNS configuration

STEP PASSED

STEP 2 - Check that DUT returned DNSInformation

STEP PASSED

STEP 3 - Validate DNS information

STEP PASSED

TEST PASSED

DEVICE-2-1-5-v14.12 SET DNS CONFIGURATION - SEARCHDOMAIN

TestResult

STEP 1 - Get DNS configuration

STEP PASSED

STEP 2 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 3 - Set DNS configuration

STEP PASSED

STEP 4 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 5 - Get DNS configuration

STEP PASSED

STEP 6 - Check that DNS configuration returned from the DUT

STEP PASSED

STEP 7 - Check that FromDHCP is false

STEP PASSED

STEP 8 - Check that the DUT returned Search Domains

STEP PASSED

STEP 9 - Validate SearchDomain value

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-2-1-6-v21.06 SET DNS CONFIGURATION - DNSMANUAL IPV4

TestResult

STEP 1 - Get DNS configuration

STEP PASSED

STEP 2 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 3 - Get network interfaces

STEP PASSED

STEP 4 - Check if DHCP must be turned off

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-2-1-8-v21.06 SET DNS CONFIGURATION - FROMDHCP

TestResult

STEP 1 - Get DNS configuration

STEP PASSED

STEP 2 - Check that valid DNS configuration returned from the DUT

STEP PASSED

STEP 3 - Get network interfaces

STEP PASSED

STEP 4 - Check if DHCP must be turned on

STEP PASSED

STEP 5 - Waiting for Hello message...

STEP PASSED

STEP 6 - Set network interface

STEP PASSED

STEP 7 - Send System Reboot message

STEP PASSED

STEP 8 - Waiting for Hello message from the DUT

STEP PASSED

STEP 9 - 5 seconds timeout after Hello

STEP PASSED

STEP 10 - Verifying Hello message

STEP PASSED

STEP 11 - Identifying right address

STEP PASSED

STEP 12 - Set DNS configuration

STEP PASSED

STEP 13 - Wait 10.000 seconds to allow the DUT to interact with DHCP server

STEP PASSED

STEP 14 - Get DNS configuration

STEP PASSED

STEP 15 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 16 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 17 - Check current DNS configuration

STEP PASSED

STEP 18 - Restore DNS configuration

STEP PASSED

STEP 19 - Waiting for Hello message...

STEP PASSED

STEP 20 - Restore network settings

STEP PASSED

STEP 21 - Send System Reboot message

STEP PASSED

STEP 22 - Waiting for Hello message from the DUT

STEP PASSED

STEP 23 - 5 seconds timeout after Hello

STEP PASSED

STEP 24 - Verifying Hello message

STEP PASSED

STEP 25 - Identifying right address

STEP PASSED

TEST PASSED

DEVICE-2-1-11-v20.12 GET NTP CONFIGURATION

TestResult

STEP 1 - Get NTP information

STEP PASSED

STEP 2 - Check that DUT returned NTP information

STEP PASSED

STEP 3 - Validate NTP information

STEP PASSED

TEST PASSED

DEVICE-2-1-12-v21.06 SET NTP CONFIGURATION - NTPMANUAL IPV4

TestResult

STEP 1 - Get NTP information

STEP PASSED

STEP 2 - Check that DUT returned NTP information

STEP PASSED

STEP 3 - Get network interfaces

STEP PASSED

STEP 4 - Check if DHCP must be turned off

STEP PASSED

STEP 5 - Set NTP configuration

STEP PASSED

STEP 6 - Get NTP information

STEP PASSED

STEP 7 - Check that DUT returned NTP information

STEP PASSED

STEP 8 - Validate current NTP configuration

STEP PASSED

STEP 9 - Restore NTP configuration

STEP PASSED

TEST PASSED

DEVICE-2-1-14-v21.06 SET NTP CONFIGURATION - FROMDHCP

TestResult

STEP 1 - Get NTP information

STEP PASSED

STEP 2 - Check that original NTP configuration returned from the DUT

STEP PASSED

STEP 3 - Get network interfaces

STEP PASSED

STEP 4 - Check if DHCP must be turned on

STEP PASSED

STEP 5 - Waiting for Hello message...

STEP PASSED

STEP 6 - Set network interface

STEP PASSED

STEP 7 - Send System Reboot message

STEP PASSED

STEP 8 - Waiting for Hello message from the DUT

STEP PASSED

STEP 9 - 5 seconds timeout after Hello

STEP PASSED

STEP 10 - Verifying Hello message

STEP PASSED

STEP 11 - Identifying right address

STEP PASSED

STEP 12 - Set NTP configuration

STEP PASSED

STEP 13 - Wait 10.000 seconds to allow the DUT to interact with DHCP server

STEP PASSED

STEP 14 - Get NTP information

STEP PASSED

STEP 15 - Check that current NTP configuration returned from the DUT

STEP PASSED

STEP 16 - Check current NTP configuration

STEP PASSED

STEP 17 - Waiting for Hello message...

STEP PASSED

STEP 18 - Restore network settings

STEP PASSED

STEP 19 - Send System Reboot message

STEP PASSED

STEP 20 - Waiting for Hello message from the DUT

STEP PASSED

STEP 21 - 5 seconds timeout after Hello

STEP PASSED

STEP 22 - Verifying Hello message

STEP PASSED

STEP 23 - Identifying right address

STEP PASSED

STEP 24 - Restore NTP configuration

STEP PASSED

TEST PASSED

DEVICE-2-1-17-v20.12 GET NETWORK INTERFACE CONFIGURATION

TestResult

STEP 1 - Get network interfaces

STEP PASSED

STEP 2 - Check if Network Interfaces returned from the DUT

STEP PASSED

TEST PASSED

DEVICE-2-1-18-v21.06 SET NETWORK INTERFACE CONFIGURATION - IPV4

TestResult

STEP 1 - Get network interfaces

STEP PASSED

STEP 2 - Check that the DUT returned current interfaces

STEP PASSED

STEP 3 - Verifying IPv4 presence

STEP PASSED

STEP 4 - Waiting for Hello message...

STEP PASSED

STEP 5 - Set network interface

STEP PASSED

STEP 6 - Send System Reboot message

STEP PASSED

STEP 7 - Waiting for Hello message from the DUT

STEP PASSED

STEP 8 - 5 seconds timeout after Hello

STEP PASSED

STEP 9 - Verifying Hello message

STEP PASSED

STEP 10 - Identifying right address

STEP PASSED

STEP 11 - Get network interfaces

STEP PASSED

STEP 12 - Verifying appliance of IPv4 static settings

STEP PASSED

STEP 13 - Waiting for Hello message...

STEP PASSED

STEP 14 - Restore network settings

STEP PASSED

STEP 15 - Send System Reboot message

STEP PASSED

STEP 16 - Waiting for Hello message from the DUT

STEP PASSED

STEP 17 - 5 seconds timeout after Hello

STEP PASSED

STEP 18 - Waiting for Hello message...

STEP PASSED

STEP 19 - Waiting for Hello message from the DUT

STEP 20 - Verifying Hello message

STEP PASSED

STEP 21 - Identifying right address

STEP PASSED

TEST PASSED

DEVICE-2-1-25-v20.12 GET NETWORK DEFAULT GATEWAY CONFIGURATION

TestResult

STEP 1 - Get Network Default Gateway

STEP PASSED

STEP 2 - Check if network default configuration returned

STEP PASSED

STEP 3 - Validate addresses

STEP PASSED

TEST PASSED

DEVICE-2-1-30-v21.06 SET NETWORK DEFAULT GATEWAY CONFIGURATION - IPV4

TestResult

STEP 1 - Get Network Default Gateway

STEP PASSED

STEP 2 - Check if original network default configuration returned

STEP PASSED

STEP 3 - Get network interfaces

STEP PASSED

STEP 4 - Check that the DUT returned current interfaces

STEP PASSED

STEP 5 - Set Network Default Gateway

STEP PASSED

STEP 6 - Wait 10 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get Network Default Gateway

STEP PASSED

STEP 8 - Check if IP address 10.1.1.1 is present in the list

STEP PASSED

STEP 9 - Set Network Default Gateway

STEP PASSED

STEP 10 - Wait 10 seconds to allow the DUT to apply settings

STEP PASSED

TEST PASSED

DEVICE-2-1-32-v20.12 NETWORK COMMAND SETHOSTNAME TEST

TestResult

STEP 1 - Get Hostname

STEP PASSED

STEP 2 - Check that the DUT returned current hostname

STEP PASSED

STEP 3 - Set Hostname

STEP PASSED

STEP 4 - Get Hostname

STEP PASSED

STEP 5 - Check that the DUT returned current hostname

STEP PASSED

STEP 6 - Verify that hostname has been changed

STEP PASSED

STEP 7 - Verify that FromDHCP is false

STEP PASSED

STEP 8 - Restore hostname

STEP PASSED

TEST PASSED

DEVICE-2-1-33-v20.12 GET NETWORK PROTOCOLS CONFIGURATION

TestResult

STEP 1 - Get Network Protocols

STEP PASSED

STEP 2 - Check if network protocols returned from the DUT

STEP PASSED

STEP 3 - Check if HTTP is present in the list

STEP PASSED

TEST PASSED

DEVICE-2-1-34-v20.12 SET NETWORK PROTOCOLS CONFIGURATION

TestResult

STEP 1 - Get Network Protocols

STEP PASSED

STEP 2 - Check if network protocols returned from the DUT

STEP PASSED

STEP 3 - Set Network Protocols

STEP PASSED

STEP 4 - Get Network Protocols

STEP PASSED

STEP 5 - Check if network protocols returned from the DUT

STEP PASSED

STEP 6 - Validating protocols

STEP PASSED

STEP 7 - Set Network Protocols

STEP PASSED

STEP 8 - Get Network Protocols

STEP PASSED

STEP 9 - Check if network protocols returned from the DUT

STEP PASSED

STEP 10 - Validating protocols

STEP PASSED

STEP 11 - Set Network Protocols

STEP PASSED

TEST PASSED

DEVICE-2-1-35-v20.12 SET NETWORK PROTOCOLS CONFIGURATION - UNSUPPORTED PROTOCOLS

TestResult

STEP 1 - Get Network Protocols

STEP PASSED

STEP 2 - Check if network protocols returned from the DUT

STEP PASSED

STEP 3 - Set Network Protocols - negative test

STEP PASSED

STEP 4 - Get Network Protocols

STEP PASSED

STEP 5 - Check if network protocols returned from the DUT

STEP PASSED

STEP 6 - Check network protocol's configurations

STEP PASSED

TEST PASSED

DEVICE-2-1-36-v20.12 GET DYNAMIC DNS CONFIGURATION

TestResult

STEP 1 - Get Dynamic DNS configuration

STEP PASSED

TEST PASSED

DEVICE-3-1-1-v14.12 SYSTEM COMMAND GETSYSTEMDATEANDTIME

TestResult

STEP 1 - Get system date and time

STEP PASSED

STEP 2 - Check that DUT returned date and time settings

STEP PASSED

STEP 3 - Validate TimeZone string

STEP PASSED

STEP 4 - Check if settings are self-consistent

STEP PASSED

STEP 5 - Validate LocalDateTime

STEP PASSED

STEP 6 - Validate UTCDateTime

STEP PASSED

TEST PASSED

DEVICE-3-1-4-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST FOR INVALID TIMEZONE

TestResult

STEP 1 - Get system date and time

STEP PASSED

STEP 2 - Set system date and time - negative test

STEP PASSED

STEP 3 - Get system date and time

STEP PASSED

STEP 4 - Check that DUT returned date and time settings

STEP PASSED

STEP 5 - Check that DUT returned TimeZone settings

STEP PASSED

STEP 6 - Check if settings are self-consistent

STEP PASSED

STEP 7 - Validate LocalDateTime

STEP PASSED

STEP 8 - Validate UTCDateTime

STEP PASSED

STEP 9 - Synchronize time

STEP PASSED

TEST PASSED

DEVICE-3-1-5-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME TEST FOR INVALID DATE

TestResult

STEP 1 - Get system date and time

STEP PASSED

STEP 2 - Set system date and time - negative test

STEP PASSED

STEP 3 - Get system date and time

STEP PASSED

STEP 4 - Check that DUT returned date and time settings

STEP PASSED

STEP 5 - Validate TimeZone string

STEP PASSED

STEP 6 - Check if settings are self-consistent

STEP PASSED

STEP 7 - Validate LocalDateTime

STEP PASSED

STEP 8 - Validate UTCDateTime

STEP PASSED

STEP 9 - Synchronize time

STEP PASSED

TEST PASSED

DEVICE-3-1-7-v21.06 SYSTEM COMMAND FACTORY DEFAULT SOFT

TestResult

STEP 1 - Set System Factory Default

STEP PASSED

STEP 2 - Wait until Reboot Timeout expires (5.000 sec)

STEP PASSED

STEP 3 - Transmit multicast PROBE message

STEP PASSED

STEP 4 - Check that answer has been received

STEP PASSED

TEST PASSED

DEVICE-3-1-8-v21.06 SYSTEM COMMAND REBOOT

TestResult

STEP 1 - Send System Reboot message

STEP PASSED

STEP 2 - Waiting for Hello message...

STEP PASSED

STEP 3 - Waiting for Hello message from the DUT

STEP PASSED

STEP 4 - 5 seconds timeout after Hello

STEP PASSED

STEP 5 - Probe device

STEP PASSED

STEP 6 - Validate probe match

STEP PASSED

TEST PASSED

DEVICE-3-1-9-v14.12 SYSTEM COMMAND DEVICE INFORMATION

TestResult

STEP 1 - Get device information

STEP PASSED

STEP 2 - Check Manufacturer information

STEP PASSED

STEP 3 - Check Model information

STEP PASSED

STEP 4 - Check FirmwareVersion information

STEP PASSED

STEP 5 - Check SerialNumber information

STEP PASSED

STEP 6 - Check HardwareId information

STEP PASSED

TEST PASSED

DEVICE-3-1-10-v14.12 SYSTEM COMMAND GETSYSTEMLOG

TestResult

STEP 1 - Get system log (system)

STEP PASSED

STEP 2 - Get system log (access)

STEP PASSED

TEST PASSED

DEVICE-3-1-11-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME

TestResult

STEP 1 - Get system date and time

STEP PASSED

STEP 2 - Set system date and time

STEP PASSED

STEP 3 - Get system date and time

STEP PASSED

STEP 4 - Check that DUT returned date and time settings

STEP PASSED

STEP 5 - Check that DateTimeType has been set.

STEP PASSED

STEP 6 - Check that DaylightSavings has been set.

STEP PASSED

STEP 7 - Check if settings are self-consistent

STEP PASSED

STEP 8 - Validate LocalDateTime

STEP PASSED

STEP 9 - Validate UTCDateTime

STEP PASSED

STEP 10 - Synchronize time

STEP PASSED

TEST PASSED

DEVICE-3-1-12-v21.06 SYSTEM COMMAND SETSYSTEMDATEANDTIME USING NTP

TestResult

STEP 1 - Get system date and time

STEP PASSED

STEP 2 - Get NTP information

STEP PASSED

STEP 3 - Set NTP configuration

STEP PASSED

STEP 4 - Set system date and time

STEP PASSED

STEP 5 - Get system date and time

STEP PASSED

STEP 6 - Check that DUT returned date and time settings

STEP PASSED

STEP 7 - Check that DateTimeType has been set.

STEP PASSED

STEP 8 - Check that DaylightSavings has been set.

STEP PASSED

STEP 9 - Check that DUT returned TimeZone settings

STEP PASSED

STEP 10 - Validate TimeZone

STEP PASSED

STEP 11 - Validate LocalDateTime

STEP PASSED

STEP 12 - Validate UTCDateTime

STEP PASSED

STEP 13 - Synchronize time

STEP PASSED

STEP 14 - Set NTP configuration

STEP PASSED

TEST PASSED

DEVICE-3-1-13-v20.06 GET SYSTEM URIS

TestResult

STEP 1 - Get service capabilities

STEP PASSED

STEP 2 - Check capabilities is returned

STEP PASSED

STEP 3 - Get System URI's

STEP PASSED

STEP 4 - Check there are non-empty System Log URIs

STEP PASSED

STEP 5 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/SystemLog'

STEP PASSED

STEP 6 - Check HTTP status code

STEP PASSED

STEP 7 - Check System Log content is returned

STEP PASSED

STEP 8 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/AccessLog'

STEP PASSED

STEP 9 - Check HTTP status code

STEP PASSED

STEP 10 - Check System Log content is returned

STEP PASSED

STEP 11 - Check Support Info URI isn't empty

STEP PASSED

STEP 12 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/SupportInfo'

STEP PASSED

STEP 13 - Check HTTP status code

STEP PASSED

STEP 14 - Check Support Info content is returned

STEP PASSED

STEP 15 - Check System Backup URI isn't empty

STEP PASSED

STEP 16 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/SystemBackup'

STEP PASSED

STEP 17 - Check HTTP status code

STEP PASSED

STEP 18 - Check System Backup content is returned

STEP PASSED

TEST PASSED

DEVICE-3-1-14-v21.06 START SYSTEM RESTORE

TestResult

STEP 1 - Get System URI's

STEP PASSED

STEP 2 - Check System Backup URI isn't empty

STEP PASSED

STEP 3 - Invoke HTTP GET request on URI 'http://192.168.3.36:8000/SystemBackup'

STEP PASSED

STEP 4 - Check HTTP status code

STEP PASSED

STEP 5 - Check System Backup content is returned

STEP PASSED

STEP 6 - Start System Restore

STEP PASSED

STEP 7 - Invoke HTTP POST request on URI 'http://192.168.3.36:8000/SystemRestore'

STEP PASSED

STEP 8 - Check HTTP status code

STEP PASSED

STEP 9 - Waiting for Hello message...

STEP PASSED

STEP 10 - Waiting for Hello message from the DUT

STEP PASSED

STEP 11 - 5 seconds timeout after Hello

STEP PASSED

STEP 12 - Probe device

STEP PASSED

TEST PASSED

DEVICE-3-1-15-v21.06 START SYSTEM RESTORE – INVALID BACKUP FILE

TestResult

STEP 1 - Start System Restore

STEP PASSED

STEP 2 - Invoke HTTP POST request on URI 'http://192.168.3.36:8000/SystemRestore'

STEP PASSED

STEP 3 - Check HTTP status code

STEP PASSED

STEP 4 - 5 seconds timeout

STEP PASSED

STEP 5 - Probe device

STEP PASSED

TEST PASSED

DEVICE-4-1-1-v20.12 SECURITY COMMAND GETUSERS

TestResult

STEP 1 - Get Users

STEP PASSED

STEP 2 - Validate response received

STEP PASSED

TEST PASSED

DEVICE-4-1-3-v20.12 SECURITY COMMAND CREATEUSERS ERROR CASE

TestResult

STEP 1 - Create users

STEP PASSED

STEP 2 - Get Users

STEP PASSED

STEP 3 - Check if the DUT returned users list

STEP PASSED

STEP 4 - Check if newly created user is present in the list

STEP PASSED

STEP 5 - Check if user has been created correctly

STEP PASSED

STEP 6 - Create User - Negative test

STEP PASSED

STEP 7 - Create User - Negative test

STEP PASSED

STEP 8 - Get Users

STEP PASSED

STEP 9 - Check if the DUT returned users list

STEP PASSED

STEP 10 - Check if no new users have been created

STEP PASSED

STEP 11 - Check if previously created user is present in the list

STEP PASSED

STEP 12 - Check if previously created user has correct level

STEP PASSED

STEP 13 - Delete users

STEP PASSED

TEST PASSED

DEVICE-4-1-4-v20.12 SECURITY COMMAND DELETEUSERS

TestResult

STEP 1 - Create users

STEP PASSED

STEP 2 - Get Users

STEP PASSED

STEP 3 - Check if the DUT returned users list

STEP PASSED

STEP 4 - Check condition

STEP PASSED

STEP 5 - Delete users

STEP PASSED

STEP 6 - Get Users

STEP PASSED

STEP 7 - Check if the DUT returned users list

STEP PASSED

STEP 8 - Check if the user has been deleted

STEP PASSED

STEP 9 - Delete users

STEP PASSED

STEP 10 - Get Users

STEP PASSED

STEP 11 - Check if the DUT returned users list

STEP PASSED

STEP 12 - Check if both users have been deleted

STEP PASSED

TEST PASSED

DEVICE-4-1-5-v20.12 SECURITY COMMAND DELETEUSERS ERROR CASE

TestResult

STEP 1 - Create users

STEP PASSED

STEP 2 - Delete Users - negative test

STEP PASSED

STEP 3 - Get Users

STEP PASSED

STEP 4 - Check if the DUT returned users list

STEP PASSED

STEP 5 - Check that the user OnvifTest1 has not been deleted

STEP PASSED

STEP 6 - Delete users

STEP PASSED

STEP 7 - Get Users

STEP PASSED

STEP 8 - Check if the DUT returned users list

STEP PASSED

STEP 9 - Check that the user OnvifTest1 has been deleted

STEP PASSED

TEST PASSED

DEVICE-4-1-7-v20.12 SECURITY COMMAND SETUSER

TestResult

STEP 1 - Create users

STEP PASSED

STEP 2 - Get Users

STEP PASSED

STEP 3 - Check if the DUT returned users list

STEP PASSED

STEP 4 - Set users

STEP PASSED

STEP 5 - Get Users

STEP PASSED

STEP 6 - Check if the DUT returned users list

STEP PASSED

STEP 7 - Check if the DUT returned modified users

STEP PASSED

STEP 8 - Set users

STEP PASSED

STEP 9 - Get Users

STEP PASSED

STEP 10 - Check if the DUT returned users list

STEP PASSED

STEP 11 - Check if the users have been modified correctly

STEP PASSED

STEP 12 - Delete users

STEP PASSED

TEST PASSED

DEVICE-4-1-8-v20.12 SECURITY COMMAND USER MANAGEMENT ERROR CASE

TestResult

STEP 1 - Create users

STEP PASSED

STEP 2 - Get Users

STEP PASSED

STEP 3 - Check if the DUT returned users list

STEP PASSED

STEP 4 - Set Users - negative test

STEP PASSED

STEP 5 - Get Users

STEP PASSED

STEP 6 - Check if the DUT returned users list

STEP PASSED

STEP 7 - Check if the user has not been modified

STEP PASSED

STEP 8 - Delete users

STEP PASSED

STEP 9 - Get Users

STEP PASSED

STEP 10 - Check if the DUT returned users list

STEP PASSED

TEST PASSED

DEVICE-4-1-9-v20.12 SECURITY COMMAND CREATEUSERS

TestResult

STEP 1 - Get Users

STEP PASSED

STEP 2 - Create users

STEP PASSED

STEP 3 - Get Users

STEP PASSED

STEP 4 - Check new user is created

STEP PASSED

STEP 5 - Create users

STEP PASSED

STEP 6 - Get Users

STEP PASSED

STEP 7 - Check new user is created

STEP PASSED

STEP 8 - Check new user is created

STEP PASSED

STEP 9 - Delete users

STEP PASSED

STEP 10 - Create users

STEP PASSED

STEP 11 - Get Users

STEP PASSED

STEP 12 - Check new user is created

STEP PASSED

STEP 13 - Delete users

STEP PASSED

STEP 14 - Check if a user with any parameters has been created

STEP PASSED

TEST PASSED

DEVICE-4-1-10-v14.12 GET REMOTE USER

TestResult

STEP 1 - Get Remote User

STEP PASSED

STEP 2 - Validating received response to GetRemoteUser command

STEP PASSED

TEST PASSED

DEVICE-4-1-11-v14.12 SET REMOTE USER

TestResult

STEP 1 - Set Remote User

STEP PASSED

STEP 2 - Get Remote User

STEP PASSED

STEP 3 - Validating received response to GetRemoteUser command

STEP PASSED

STEP 4 - Set Remote User

STEP PASSED

STEP 5 - Get Remote User

STEP PASSED

STEP 6 - Validating received response to GetRemoteUser command

STEP PASSED

STEP 7 - Set Remote User

STEP PASSED

STEP 8 - Get Remote User

STEP PASSED

STEP 9 - Validating received response to GetRemoteUser command

STEP PASSED

TEST PASSED

DEVICE-5-1-1-v16.07 IO COMMAND GETRELAYOUTPUTS

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

TEST PASSED

DEVICE-5-1-2-v16.07 RELAY OUTPUTS COUNT IN GETRELAYOUTPUTS AND GETCAPABILITIES

TestResult

STEP 1 - Get capabilities

STEP PASSED

STEP 2 - Check that DUT returned capabilities

STEP PASSED

STEP 3 - Check that DUT returned device capabilities

STEP PASSED

STEP 4 - Check that IO capabilities returned

STEP PASSED

STEP 5 - Get relay outputs

STEP PASSED

STEP 6 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 7 - Check that count of relay outputs is the same

STEP PASSED

TEST PASSED

DEVICE-5-1-3-v16.07 IO COMMAND SETRELAYOUTPUTSETTINGS

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = open, Mode = Bistable)

STEP PASSED

STEP 4 - Get relay outputs

STEP PASSED

STEP 5 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 6 - Find current output settings

STEP PASSED

STEP 7 - Compare expected and actual relay output properties

STEP PASSED

STEP 8 - Set relay output settings (IdleState = closed, Mode = Bistable)

STEP PASSED

STEP 9 - Get relay outputs

STEP PASSED

STEP 10 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 11 - Find current output settings

STEP PASSED

STEP 12 - Compare expected and actual relay output properties

STEP PASSED

STEP 13 - Set relay output settings (IdleState = open, Mode = Monostable)

STEP PASSED

STEP 14 - Get relay outputs

STEP PASSED

STEP 15 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 16 - Find current output settings

STEP PASSED

STEP 17 - Compare expected and actual relay output properties

STEP PASSED

STEP 18 - Set relay output settings (IdleState = closed, Mode = Monostable)

STEP PASSED

STEP 19 - Get relay outputs

STEP PASSED

STEP 20 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 21 - Find current output settings

STEP PASSED

STEP 22 - Compare expected and actual relay output properties

STEP PASSED

TEST PASSED

DEVICE-5-1-5-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – BISTABLE MODE (OPENED IDLE STATE)

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = open, Mode = Bistable)

STEP PASSED

STEP 4 - Set relay output state

STEP PASSED

STEP 5 - Set relay output state

STEP PASSED

STEP 6 - Restore output settings

STEP PASSED

TEST PASSED

DEVICE-5-1-6-v14.12 IO COMMAND SETRELAYOUTPUTSTATE – BISTABLE MODE (CLOSED IDLE STATE)

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = closed, Mode = Bistable)

STEP PASSED

STEP 4 - Set relay output state

STEP PASSED

STEP 5 - Set relay output state

STEP PASSED

STEP 6 - Restore output settings

STEP PASSED

TEST PASSED

DEVICE-5-1-7-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(OPENED IDLE STATE)

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = open, Mode = Monostable)

STEP PASSED

STEP 4 - Set relay output state

STEP PASSED

STEP 5 - Wait 20 seconds

STEP PASSED

STEP 6 - Restore output settings

STEP PASSED

TEST PASSED

DEVICE-5-1-8-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(CLOSED IDLE STATE)

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = closed, Mode = Monostable)

STEP PASSED

STEP 4 - Set relay output state

STEP PASSED

STEP 5 - Wait 20 seconds

STEP PASSED

STEP 6 - Restore output settings

STEP PASSED

TEST PASSED

DEVICE-5-1-9-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – MONOSTABLE MODE
(INACTIVE BEFORE DELAYTIME EXPIRED)

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings (IdleState = open, Mode = Monostable)

STEP PASSED

STEP 4 - Set relay output state

STEP PASSED

STEP 5 - Check if timeout has not expired

STEP PASSED

STEP 6 - Set relay output state

STEP PASSED

STEP 7 - Check if timeout has not expired

STEP PASSED

STEP 8 - Check if timeout expired

STEP PASSED

STEP 9 - Set relay output settings (IdleState = closed, Mode = Monostable)

STEP PASSED

STEP 10 - Set relay output state

STEP PASSED

STEP 11 - Check if timeout has not expired

STEP PASSED

STEP 12 - Set relay output state

STEP PASSED

STEP 13 - Check if timeout has not expired

STEP PASSED

STEP 14 - Check if timeout expired

STEP PASSED

STEP 15 - Restore output settings

STEP PASSED

TEST PASSED

DEVICE-5-1-11-v16.07 IO COMMAND SETRELAYOUTPUTSETTINGS – INVALID TOKEN

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings - negative test

STEP PASSED

TEST PASSED

DEVICE-5-1-12-v16.07 IO COMMAND SETRELAYOUTPUTSTATE – INVALID TOKEN

TestResult

STEP 1 - Get relay outputs

STEP PASSED

STEP 2 - Check that the DUT sent relay outputs information

STEP PASSED

STEP 3 - Set relay output settings - negative test

STEP PASSED

TEST PASSED

DEVICE-6-1-1-v21.06 DEVICE MANAGEMENT - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get DNS configuration

STEP PASSED

STEP 4 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-6-1-2-v21.06 DEVICE MANAGEMENT - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get DNS configuration

STEP PASSED

STEP 4 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-6-1-3-v21.06 DEVICE MANAGEMENT - NAMESPACES (NOT STANDARD PREFIXES)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get DNS configuration

STEP PASSED

STEP 4 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-6-1-4-v21.06 DEVICE MANAGEMENT - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get DNS configuration

STEP PASSED

STEP 4 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-6-1-5-v21.06 DEVICE MANAGEMENT - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get DNS configuration

STEP PASSED

STEP 4 - Check that original DNS configuration returned from the DUT

STEP PASSED

STEP 5 - Set DNS configuration

STEP PASSED

STEP 6 - Wait 10.000 seconds to allow the DUT to apply settings

STEP PASSED

STEP 7 - Get DNS configuration

STEP PASSED

STEP 8 - Check that current DNS configuration returned from the DUT

STEP PASSED

STEP 9 - Check current DNS configuration

STEP PASSED

STEP 10 - Restore DNS configuration

STEP PASSED

TEST PASSED

DEVICE-8-1-1-v17.01 AUXILIARY COMMANDS

TestResult

STEP 1 - Get service capabilities

STEP PASSED

TEST PASSED

Event Handling

EVENT-1-1-2-v19.06 GET EVENT PROPERTIES

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

STEP 3 - Check that the DUT returned Topic Expression Dialects

STEP PASSED

STEP 4 - Check that Mandatory Topic Expression Dialect <http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete> is supported

STEP PASSED

STEP 5 - Check that Mandatory Topic Expression Dialect <http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet> is supported

STEP PASSED

STEP 6 - Check that the DUT returned Message Content Filter Dialects

STEP PASSED

STEP 7 - Check if the DUT supports mandatory Message Content Filter Dialect

<http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter>

STEP PASSED

STEP 8 - Check if response contains at least one topic namespace and that it is a valid string for an uri

STEP PASSED

STEP 9 - Check that the TopicSet returned is not null

STEP PASSED

STEP 10 - Check that the DUT returned not empty TopicSet

STEP PASSED

TEST PASSED

EVENT-2-1-9-v14.12 BASIC NOTIFICATION INTERFACE - SUBSCRIBE

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Creating listening server

STEP PASSED

STEP 3 - Send Subscribe request

STEP PASSED

STEP 4 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 5 - Check that CurrentTime is specified

STEP PASSED

STEP 6 - Check that TerminationTime is specified

STEP PASSED

STEP 7 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 8 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 9 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 10 - Check if SubscriptionReference contains address

STEP PASSED

STEP 11 - Check that URL specified is valid

STEP PASSED

STEP 12 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-12-v14.12 BASIC NOTIFICATION INTERFACE - RENEW

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Creating listening server

STEP PASSED

STEP 3 - Send Subscribe request

STEP PASSED

STEP 4 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 5 - Check that CurrentTime is specified

STEP PASSED

STEP 6 - Check that TerminationTime is specified

STEP PASSED

STEP 7 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 8 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 9 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 10 - Check if SubscriptionReference contains address

STEP PASSED

STEP 11 - Check that URL specified is valid

STEP PASSED

STEP 12 - Renew subscription

STEP PASSED

STEP 13 - Renew subscription

STEP PASSED

STEP 14 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-17-v14.12 BASIC NOTIFICATION INTERFACE - NOTIFY

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

Timeout of 60 seconds will be used

STEP 3 - Creating listening server

STEP PASSED

STEP 4 - Send Subscribe request

STEP PASSED

STEP 5 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 6 - Check that CurrentTime is specified

STEP PASSED

STEP 7 - Check that TerminationTime is specified

STEP PASSED

STEP 8 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 9 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 10 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 11 - Check if SubscriptionReference contains address

STEP PASSED

STEP 12 - Check that URL specified is valid

STEP PASSED

STEP 13 - Set Synchronization Point

STEP PASSED

STEP 14 - Wait for notification

STEP PASSED

STEP 15 - Receiving notification

STEP PASSED

STEP 16 - Receiving notification

STEP PASSED

STEP 17 - Validate notifications SOAP packet

STEP PASSED

STEP 18 - Validate Headers

STEP PASSED

STEP 19 - Validate notifications SOAP packet

STEP PASSED

STEP 20 - Validate Headers

STEP PASSED

STEP 21 - Check that DUT sent notification messages

STEP PASSED

STEP 22 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 23 - Validate messages

STEP PASSED

STEP 24 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-18-v14.12 BASIC NOTIFICATION INTERFACE - NOTIFY FILTER

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

Timeout of 60 seconds will be used

STEP 3 - Parse topic

STEP PASSED

STEP 4 - Creating listening server

STEP PASSED

STEP 5 - Send Subscribe request

STEP PASSED

STEP 6 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 7 - Check that CurrentTime is specified

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Set Synchronization Point

STEP PASSED

STEP 15 - Wait for notification

STEP PASSED

STEP 16 - Receiving notification

STEP PASSED

STEP 17 - Receiving notification

STEP PASSED

STEP 18 - Validate notifications SOAP packet

STEP PASSED

STEP 19 - Validate Headers

STEP PASSED

STEP 20 - Validate notifications SOAP packet

STEP PASSED

STEP 21 - Validate Headers

STEP PASSED

STEP 22 - Check that DUT sent notification messages

STEP PASSED

STEP 23 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 24 - Validate messages

STEP PASSED

STEP 25 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-24-v17.06 BASIC NOTIFICATION INTERFACE - SET SYNCHRONIZATION POINT

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Creating listening server

STEP PASSED

STEP 8 - Send Subscribe request

STEP PASSED

STEP 9 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 10 - Check that CurrentTime is specified

STEP PASSED

STEP 11 - Check that TerminationTime is specified

STEP PASSED

STEP 12 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 13 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 14 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 15 - Check if SubscriptionReference contains address

STEP PASSED

STEP 16 - Check that URL specified is valid

STEP PASSED

STEP 17 - Wait for notification

STEP PASSED

STEP 18 - Receiving notification

STEP PASSED

STEP 19 - Validate notifications SOAP packet

STEP PASSED

STEP 20 - Validate Headers

STEP PASSED

STEP 21 - Check that DUT sent notification messages

STEP PASSED

STEP 22 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 23 - Checking received notification matches to the topic specified on Management tab

STEP PASSED

STEP 24 - Set Synchronization Point

STEP PASSED

STEP 25 - Wait for notification

STEP PASSED

STEP 26 - Receiving notification

STEP PASSED

STEP 27 - Validate notifications SOAP packet

STEP PASSED

STEP 28 - Validate Headers

STEP PASSED

STEP 29 - Check that DUT sent notification messages

STEP PASSED

STEP 30 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 31 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-25-v17.06 BASIC NOTIFICATION INTERFACE – CONJUNCTION IN NOTIFY FILTER (OR OPERATION)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Creating listening server

STEP PASSED

STEP 9 - Send Subscribe request

STEP PASSED

STEP 10 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 11 - Check that CurrentTime is specified

STEP PASSED

STEP 12 - Check that TerminationTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Wait for notification

STEP PASSED

STEP 19 - Receiving notification

STEP PASSED

STEP 20 - Receiving notification

STEP PASSED

STEP 21 - Validate notifications SOAP packet

STEP PASSED

STEP 22 - Validate Headers

STEP PASSED

STEP 23 - Validate notifications SOAP packet

STEP PASSED

STEP 24 - Validate Headers

STEP PASSED

STEP 25 - Check that DUT sent notification messages

STEP PASSED

STEP 26 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 27 - Check if the DUT returned only required notifications

STEP PASSED

STEP 28 - Check that DUT sent notification messages

STEP PASSED

STEP 29 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 30 - Check if the DUT returned only required notifications

STEP PASSED

STEP 31 - Check if the DUT returned all required notifications

STEP PASSED

STEP 32 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-26-v17.06 BASIC NOTIFICATION INTERFACE – TOPIC SUB-TREE IN PULLMESSAGES FILTER

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Creating listening server

STEP PASSED

STEP 9 - Send Subscribe request

STEP PASSED

STEP 10 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 11 - Check that CurrentTime is specified

STEP PASSED

STEP 12 - Check that TerminationTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Wait for notification

STEP PASSED

STEP 19 - Receiving notification

STEP PASSED

STEP 20 - Receiving notification

STEP PASSED

STEP 21 - Validate notifications SOAP packet

STEP PASSED

STEP 22 - Validate Headers

STEP PASSED

STEP 23 - Validate notifications SOAP packet

STEP PASSED

STEP 24 - Validate Headers

STEP PASSED

STEP 25 - Check that DUT sent notification messages

STEP PASSED

STEP 26 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 27 - Check if the DUT returned notifications with the root element is equal to "tns1:Device//."

STEP PASSED

STEP 28 - Check if the DUT returned all required notifications

STEP PASSED

STEP 29 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-27-v17.06 BASIC NOTIFICATION INTERFACE – CONJUNCTION IN NOTIFY FILTER (TOPIC SUB-TREE AND OR OPERATION)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Creating listening server

STEP PASSED

STEP 9 - Send Subscribe request

STEP PASSED

STEP 10 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 11 - Check that CurrentTime is specified

STEP PASSED

STEP 12 - Check that TerminationTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Wait for notification

STEP PASSED

STEP 19 - Receiving notification

STEP PASSED

STEP 20 - Receiving notification

STEP PASSED

STEP 21 - Validate notifications SOAP packet

STEP PASSED

STEP 22 - Validate Headers

STEP PASSED

STEP 23 - Validate notifications SOAP packet

STEP PASSED

STEP 24 - Validate Headers

STEP PASSED

STEP 25 - Check that DUT sent notification messages

STEP PASSED

STEP 26 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 27 - Check if the DUT returned notifications with the root element is equal to "tns1:Device//." or with the topic is equal to "tns1:Device/Trigger/Relay"

STEP PASSED

STEP 28 - Check that DUT sent notification messages

STEP PASSED

STEP 29 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 30 - Check if the DUT returned notifications with the root element is equal to "tns1:Device//." or with the topic is equal to "tns1:Device/Trigger/Relay"

STEP PASSED

STEP 31 - Check if the DUT returned all required notifications

STEP PASSED

STEP 32 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-28-v17.12 BASIC NOTIFICATION INTERFACE - UNSUBSCRIBE

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Creating listening server

STEP PASSED

STEP 5 - Send Subscribe request

STEP PASSED

STEP 6 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 7 - Check that CurrentTime is specified

STEP PASSED

STEP 8 - Check that TerminationTime is specified

STEP PASSED

STEP 9 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 10 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 11 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 12 - Check if SubscriptionReference contains address

STEP PASSED

STEP 13 - Check that URL specified is valid

STEP PASSED

STEP 14 - Waiting one second

STEP PASSED

STEP 15 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-2-1-29-v18.06 BASIC NOTIFICATION INTERFACE - MESSAGE CONTENT FILTER

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Check the DUT returned at least one MessageContentFilterDialect item

STEP PASSED

STEP 8 - Parse topic

STEP PASSED

STEP 9 - Creating listening server

STEP PASSED

STEP 10 - Send Subscribe request

STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 12 - Check that CurrentTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 17 - Check if SubscriptionReference contains address

STEP PASSED

STEP 18 - Check that URL specified is valid

STEP PASSED

STEP 19 - Wait for notification

STEP PASSED

STEP 20 - Receiving notification

STEP PASSED

STEP 21 - Validate notifications SOAP packet

STEP PASSED

STEP 22 - Validate Headers

STEP PASSED

STEP 23 - Check that DUT sent notification messages

STEP PASSED

STEP 24 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 25 - Check if the DUT returned required notification

STEP PASSED

STEP 26 - Send Unsubscribe request

STEP PASSED

STEP 27 - Wait for 1 second(s) to complete the operation

STEP PASSED

STEP 28 - Check if the DUT returned notification message with 'Source.SimpleItem' element and 'Name', 'Value' attributes

STEP PASSED

STEP 29 - Creating listening server

STEP PASSED

STEP 30 - Send Subscribe request

STEP PASSED

STEP 31 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 32 - Check that CurrentTime is specified

STEP PASSED

STEP 33 - Check that TerminationTime is specified

STEP PASSED

STEP 34 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 35 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 36 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 37 - Check if SubscriptionReference contains address

STEP PASSED

STEP 38 - Check that URL specified is valid

STEP PASSED

Waiting for notification [Topic = 'tns1:Device/Trigger/DigitalInput', PropertyOperation = 'Initialized']

STEP 39 - Wait for notification

STEP PASSED

STEP 40 - Receiving notification

STEP PASSED

STEP 41 - Validate notifications SOAP packet

STEP PASSED

STEP 42 - Validate Headers

STEP PASSED

STEP 43 - Check that DUT sent notification messages

STEP PASSED

STEP 44 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 45 - Check that each returned notification message contains SimpleItem element with Name = 'InputToken' and with Value = 'DigitalInputToken_1'

STEP PASSED

STEP 46 - Check if the DUT returned required notification

STEP PASSED

STEP 47 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-9-v14.12 REALTIME PULLPOINT SUBSCRIPTION - CREATE PULL POINT SUBSCRIPTION

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Create Pull Point Subscription

STEP PASSED

STEP 3 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 4 - Check if SubscriptionReference contains address

STEP PASSED

STEP 5 - Check that URL specified is valid

STEP PASSED

STEP 6 - Check that TerminationTime is specified

STEP PASSED

STEP 7 - Validate times

STEP PASSED

STEP 8 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-3-1-15-v14.12 REALTIME PULLPOINT SUBSCRIPTION - PULLMESSAGES

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

Timeout of 60 seconds will be used

STEP 3 - Create Pull Point Subscription

STEP PASSED

STEP 4 - Check that TerminationTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - Send PullMessages request

STEP PASSED

STEP 11 - Set Synchronization Point

STEP PASSED

STEP 12 - Get PullMessages response

STEP PASSED

STEP 13 - Check that DUT sent notification messages

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check that a maximum number of 2 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 16 - Response is not empty

STEP PASSED

STEP 17 - Validate messages

STEP PASSED

STEP 18 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-3-1-16-v21.06 REALTIME PULLPOINT SUBSCRIPTION - PULLMESSAGES FILTER

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

STEP 3 - Parse topic

STEP PASSED

Timeout of 60 seconds will be used

STEP 4 - Create Pull Point Subscription

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 7 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 8 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 9 - Check if SubscriptionReference contains address

STEP PASSED

STEP 10 - Check that URL specified is valid

STEP PASSED

STEP 11 - Send PullMessages request

STEP PASSED

STEP 12 - Set Synchronization Point

STEP PASSED

STEP 13 - Get PullMessages response

STEP PASSED

STEP 14 - Check that DUT sent notification messages

STEP PASSED

STEP 15 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 16 - Check that a maximum number of 2 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 17 - Response is not empty

STEP PASSED

STEP 18 - Validate messages

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-3-1-24-v14.12 REALTIME PULLPOINT SUBSCRIPTION – PULLMESSAGES AS KEEP-ALIVE

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Get Event Properties

STEP PASSED

STEP 3 - Create Pull Point Subscription

STEP PASSED

STEP 4 - Check that TerminationTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - 1 second after CreatePullPointSubscription

STEP PASSED

STEP 11 - Validating Current Time and Termination Time in CreatePullPointSubscription response

STEP PASSED

STEP 12 - Send PullMessages request

STEP PASSED

STEP 13 - Get PullMessages response

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 16 - Response is not empty

STEP PASSED

STEP 17 - Validating Current Time and Termination Time in PullMessages response

STEP PASSED

STEP 18 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-25-v17.06 REALTIME PULLPOINT SUBSCRIPTION – SET SYNCHRONIZATION POINT

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - 1 second after CreatePullPointSubscription

STEP PASSED

STEP 16 - Send PullMessages request

STEP PASSED

STEP 17 - Check that DUT sent notification messages

STEP PASSED

STEP 18 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 19 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 20 - Response is not empty

STEP PASSED

STEP 21 - Checking received notification matches to the topic specified on Management tab

STEP PASSED

STEP 22 - Set Synchronization Point

STEP PASSED

STEP 23 - 1 second timeout

STEP PASSED

STEP 24 - Send PullMessages request

STEP PASSED

STEP 25 - Check that DUT sent notification messages

STEP PASSED

STEP 26 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 27 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 28 - Response is not empty

STEP PASSED

STEP 29 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-32-v17.06 REALTIME PULLPOINT SUBSCRIPTION – PULLMESSAGES TIMEOUT

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Create Pull Point Subscription

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 7 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 8 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 9 - Check if SubscriptionReference contains address

STEP PASSED

STEP 10 - Check that URL specified is valid

STEP PASSED

STEP 11 - Send PullMessages request

STEP PASSED

STEP 12 - Check that the termination time is greater than the current time

STEP PASSED

STEP 13 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-33-v21.06 REALTIME PULLPOINT SUBSCRIPTION – CONJUNCTION IN PULLMESSAGES FILTER (OR OPERATION)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Send PullMessages request

STEP PASSED

STEP 16 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 17 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 18 - Response is not empty

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notifications

STEP PASSED

STEP 24 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-34-v21.06 REALTIME PULLPOINT SUBSCRIPTION – TOPIC SUB-TREE IN PULLMESSAGES FILTER

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Send PullMessages request

STEP PASSED

STEP 16 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 17 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 18 - Response is not empty

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notifications

STEP PASSED

STEP 24 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-35-v21.06 REALTIME PULLPOINT SUBSCRIPTION – CONJUNCTION IN NOTIFY FILTER (TOPIC SUB-TREE AND OR OPERATION)

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Parse topic

STEP PASSED

STEP 8 - Create Pull Point Subscription

STEP PASSED

STEP 9 - Check that TerminationTime is specified

STEP PASSED

STEP 10 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 11 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 12 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 13 - Check if SubscriptionReference contains address

STEP PASSED

STEP 14 - Check that URL specified is valid

STEP PASSED

STEP 15 - Send PullMessages request

STEP PASSED

STEP 16 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 17 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 18 - Response is not empty

STEP PASSED

STEP 19 - Send PullMessages request

STEP PASSED

STEP 20 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 21 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 22 - Response is not empty

STEP PASSED

STEP 23 - Waiting for notifications

STEP PASSED

STEP 24 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-36-v17.12 REALTIME PULLPOINT SUBSCRIPTION - UNSUBSCRIBE

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Create Pull Point Subscription

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 7 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 8 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 9 - Check if SubscriptionReference contains address

STEP PASSED

STEP 10 - Check that URL specified is valid

STEP PASSED

STEP 11 - Waiting one second

STEP PASSED

STEP 12 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-37-v17.12 REALTIME PULLPOINT SUBSCRIPTION – MAXIMUM SUPPORTED NUMBER OF NOTIFICATION PULL POINTS

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Service Capabilities(Event)

STEP PASSED

STEP 7 - Check if EventServiceCapabilities item contains MaxPullPoints

STEP PASSED

STEP 8 - Get Event Properties

STEP PASSED

STEP 9 - Create Pull Point Subscription

STEP PASSED

STEP 10 - Check that TerminationTime is specified

STEP PASSED

STEP 11 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 12 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 13 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 14 - Check if SubscriptionReference contains address

STEP PASSED

STEP 15 - Check that URL specified is valid

STEP PASSED

STEP 16 - Create Pull Point Subscription

STEP PASSED

STEP 17 - Check that TerminationTime is specified

STEP PASSED

STEP 18 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 19 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 20 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 21 - Check if SubscriptionReference contains address

STEP PASSED

STEP 22 - Check that URL specified is valid

STEP PASSED

STEP 23 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 24 - Create Pull Point Subscription

STEP PASSED

STEP 25 - Check that TerminationTime is specified

STEP PASSED

STEP 26 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 27 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 28 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 29 - Check if SubscriptionReference contains address

STEP PASSED

STEP 30 - Check that URL specified is valid

STEP PASSED

STEP 31 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 32 - Create Pull Point Subscription

STEP PASSED

STEP 33 - Check that TerminationTime is specified

STEP PASSED

STEP 34 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 35 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 36 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 37 - Check if SubscriptionReference contains address

STEP PASSED

STEP 38 - Check that URL specified is valid

STEP PASSED

STEP 39 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 40 - Create Pull Point Subscription

STEP PASSED

STEP 41 - Check that TerminationTime is specified

STEP PASSED

STEP 42 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 43 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 44 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 45 - Check if SubscriptionReference contains address

STEP PASSED

STEP 46 - Check that URL specified is valid

STEP PASSED

STEP 47 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 48 - Create Pull Point Subscription

STEP PASSED

STEP 49 - Check that TerminationTime is specified

STEP PASSED

STEP 50 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 51 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 52 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 53 - Check if SubscriptionReference contains address

STEP PASSED

STEP 54 - Check that URL specified is valid

STEP PASSED

STEP 55 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 56 - Create Pull Point Subscription

STEP PASSED

STEP 57 - Check that TerminationTime is specified

STEP PASSED

STEP 58 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 59 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 60 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 61 - Check if SubscriptionReference contains address

STEP PASSED

STEP 62 - Check that URL specified is valid

STEP PASSED

STEP 63 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 64 - Create Pull Point Subscription

STEP PASSED

STEP 65 - Check that TerminationTime is specified

STEP PASSED

STEP 66 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 67 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 68 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 69 - Check if SubscriptionReference contains address

STEP PASSED

STEP 70 - Check that URL specified is valid

STEP PASSED

STEP 71 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 72 - Create Pull Point Subscription

STEP PASSED

STEP 73 - Check that TerminationTime is specified

STEP PASSED

STEP 74 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 75 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 76 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 77 - Check if SubscriptionReference contains address

STEP PASSED

STEP 78 - Check that URL specified is valid

STEP PASSED

STEP 79 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 80 - Create Pull Point Subscription

STEP PASSED

STEP 81 - Check that TerminationTime is specified

STEP PASSED

STEP 82 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 83 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 84 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 85 - Check if SubscriptionReference contains address

STEP PASSED

STEP 86 - Check that URL specified is valid

STEP PASSED

STEP 87 - Check that the DUT did not create the subscriptions with the same id

STEP PASSED

STEP 88 - Send PullMessages request

STEP PASSED

STEP 89 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 90 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 91 - Response is not empty

STEP PASSED

STEP 92 - Waiting for notification

STEP PASSED

STEP 93 - Send PullMessages request

STEP PASSED

STEP 94 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 95 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 96 - Response is not empty

STEP PASSED

STEP 97 - Waiting for notification

STEP PASSED

STEP 98 - Send PullMessages request

STEP PASSED

STEP 99 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 100 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 101 - Response is not empty

STEP PASSED

STEP 102 - Waiting for notification

STEP PASSED

STEP 103 - Send PullMessages request

STEP PASSED

STEP 104 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 105 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 106 - Response is not empty

STEP PASSED

STEP 107 - Waiting for notification

STEP PASSED

STEP 108 - Send PullMessages request

STEP PASSED

STEP 109 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 110 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 111 - Response is not empty

STEP PASSED

STEP 112 - Waiting for notification

STEP PASSED

STEP 113 - Send PullMessages request

STEP PASSED

STEP 114 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 115 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 116 - Response is not empty

STEP PASSED

STEP 117 - Waiting for notification

STEP PASSED

STEP 118 - Send PullMessages request

STEP PASSED

STEP 119 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 120 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 121 - Response is not empty

STEP PASSED

STEP 122 - Waiting for notification

STEP PASSED

STEP 123 - Send PullMessages request

STEP PASSED

STEP 124 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 125 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 126 - Response is not empty

STEP PASSED

STEP 127 - Waiting for notification

STEP PASSED

STEP 128 - Send PullMessages request

STEP PASSED

STEP 129 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 130 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 131 - Response is not empty

STEP PASSED

STEP 132 - Waiting for notification

STEP PASSED

STEP 133 - Send PullMessages request

STEP PASSED

STEP 134 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 135 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 136 - Response is not empty

STEP PASSED

STEP 137 - Waiting for notification

STEP PASSED

STEP 138 - Send Unsubscribe request

STEP PASSED

STEP 139 - Send Unsubscribe request

STEP PASSED

STEP 140 - Send Unsubscribe request

STEP PASSED

STEP 141 - Send Unsubscribe request

STEP PASSED

STEP 142 - Send Unsubscribe request

STEP PASSED

STEP 143 - Send Unsubscribe request

STEP PASSED

STEP 144 - Send Unsubscribe request

STEP PASSED

STEP 145 - Send Unsubscribe request

STEP PASSED

STEP 146 - Send Unsubscribe request

STEP PASSED

STEP 147 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-3-1-38-v18.06 REALTIME PULLPOINT SUBSCRIPTION - MESSAGE CONTENT FILTER

TestResult

STEP 1 - Get Device service address

STEP PASSED

STEP 2 - Check that the DUT returned Device service address

STEP PASSED

STEP 3 - Get Services

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Properties

STEP PASSED

STEP 7 - Check the DUT returned at least one MessageContentFilterDialect item

STEP PASSED

STEP 8 - Parse topic

STEP PASSED

STEP 9 - Create Pull Point Subscription

STEP PASSED

STEP 10 - Check that TerminationTime is specified

STEP PASSED

STEP 11 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 12 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 13 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 14 - Check if SubscriptionReference contains address

STEP PASSED

STEP 15 - Check that URL specified is valid

STEP PASSED

STEP 16 - Send PullMessages request

STEP PASSED

STEP 17 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 18 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 19 - Response is not empty

STEP PASSED

STEP 20 - Waiting for notifications [Topic = 'tns1:Device/Trigger/DigitalInput', PropertyOperation = 'Initialized']

STEP PASSED

STEP 21 - Send Unsubscribe request

STEP PASSED

STEP 22 - Wait for 1 second(s) to complete the operation

STEP PASSED

STEP 23 - Check if the DUT returned notification message with 'Source.SimpleItem' element and 'Name', 'Value' attributes

STEP PASSED

STEP 24 - Create Pull Point Subscription

STEP PASSED

STEP 25 - Check that TerminationTime is specified

STEP PASSED

STEP 26 - Check that TerminationTime and CurrentTime has reasonable values

STEP PASSED

STEP 27 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 28 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 29 - Check if SubscriptionReference contains address

STEP PASSED

STEP 30 - Check that URL specified is valid

STEP PASSED

STEP 31 - Send PullMessages request

STEP PASSED

STEP 32 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 33 - Check that a maximum number of 1 Notification Messages is included in PullMessagesResponse

STEP PASSED

STEP 34 - Response is not empty

STEP PASSED

STEP 35 - Waiting for notifications [Topic = 'tns1:Device/Trigger/DigitalInput', PropertyOperation = 'Initialized']

STEP PASSED

STEP 36 - Send Unsubscribe request

STEP PASSED

TEST PASSED

EVENT-4-1-6-v16.07 EVENT - NAMESPACES (DEFAULT NAMESPACES FOR EACH TAG)

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Send Subscribe request

STEP PASSED

STEP 3 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 4 - Check that CurrentTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - Send Subscribe request

STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 12 - Check that CurrentTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Check if reaction to request was the same

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

STEP 20 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-4-1-7-v16.07 EVENT - NAMESPACES (DEFAULT NAMESPACES FOR PARENT TAG)

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Send Subscribe request

STEP PASSED

STEP 3 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 4 - Check that CurrentTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - Send Subscribe request

STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 12 - Check that CurrentTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Check if reaction to request was the same

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

STEP 20 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-4-1-8-v16.07 EVENT - NAMESPACES (NOT STANDARD PREFIXES)

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Send Subscribe request

STEP PASSED

STEP 3 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 4 - Check that CurrentTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - Send Subscribe request

STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 12 - Check that CurrentTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Check if reaction to request was the same

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

STEP 20 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-4-1-9-v16.07 EVENT - NAMESPACES (DIFFERENT PREFIXES FOR THE SAME NAMESPACE)

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Send Subscribe request

STEP PASSED

STEP 3 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 4 - Check that CurrentTime is specified

STEP PASSED

STEP 5 - Check that TerminationTime is specified

STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 8 - Check if SubscriptionReference contains address

STEP PASSED

STEP 9 - Check that URL specified is valid

STEP PASSED

STEP 10 - Send Subscribe request

STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response

STEP PASSED

STEP 12 - Check that CurrentTime is specified

STEP PASSED

STEP 13 - Check that TerminationTime is specified

STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime

STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference

STEP PASSED

STEP 16 - Check if SubscriptionReference contains address

STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Check if reaction to request was the same

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

STEP 20 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-4-1-10-v16.07 EVENT - NAMESPACES (THE SAME PREFIX FOR DIFFERENT NAMESPACES)

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Send Subscribe request

STEP PASSED

STEP 3 - Check that the DUT returned Subscribe response
STEP PASSED

STEP 4 - Check that CurrentTime is specified
STEP PASSED

STEP 5 - Check that TerminationTime is specified
STEP PASSED

STEP 6 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 7 - Check if the DUT returned SubscriptionReference
STEP PASSED

STEP 8 - Check if SubscriptionReference contains address
STEP PASSED

STEP 9 - Check that URL specified is valid
STEP PASSED

STEP 10 - Send Subscribe request
STEP PASSED

STEP 11 - Check that the DUT returned Subscribe response
STEP PASSED

STEP 12 - Check that CurrentTime is specified
STEP PASSED

STEP 13 - Check that TerminationTime is specified
STEP PASSED

STEP 14 - Validate CurrentTime and TerminationTime
STEP PASSED

STEP 15 - Check if the DUT returned SubscriptionReference
STEP PASSED

STEP 16 - Check if SubscriptionReference contains address
STEP PASSED

STEP 17 - Check that URL specified is valid

STEP PASSED

STEP 18 - Check if reaction to request was the same

STEP PASSED

STEP 19 - Delete Subscription Manager

STEP PASSED

STEP 20 - Delete Subscription Manager

STEP PASSED

TEST PASSED

EVENT-5-1-1-v20.06 EVENT SERVICE CAPABILITIES

TestResult

STEP 1 - Get Event service address

STEP PASSED

STEP 2 - Check that the DUT returned Event service address

STEP PASSED

STEP 3 - Get Event Service Capabilities

STEP PASSED

TEST PASSED

EVENT-5-1-2-v20.06 GET SERVICES AND EVENT SERVICE CAPABILITIES CONSISTENCY

TestResult

STEP 1 - Get Services

STEP PASSED

STEP 2 - Check that the DUT returned events service information

STEP PASSED

STEP 3 - Check that the DUT returned Capabilities element

STEP PASSED

STEP 4 - Get Event service address

STEP PASSED

STEP 5 - Check that the DUT returned Event service address

STEP PASSED

STEP 6 - Get Event Service Capabilities

STEP PASSED

STEP 7 - Parse Capabilities element in GetServices response

STEP PASSED

STEP 8 - Compare Capabilities

STEP PASSED

TEST PASSED