



Device: 192.168.15.56:8080

Test time: 2019-04-17 19:54:03.869584

### 1.Test: GetVideoSources(Media)

- GetVideoSources is supported
- Test Feature: None
- Response: "[(VideoSource){\n \_token = \"h264\"\n Framerate = 30.0\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n }]"

### 2.Test: GetVideoSourceConfigurations(Media)

- GetVideoSourceConfigurations is supported
- Test Feature: None
- Response: "[(VideoSourceConfiguration){\n \_token = \"h264\"\n Name = \"h264\"\n UseCount = 3\n SourceToken = \"h264\"\n Bounds = \n (IntRectangle){\n \_y = 0\n \_width = 1920\n \_x = 0\n \_height = 1080\n }\n }]"

### 3.Test: GetVideoSourceConfigurationOptions(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 4.Test: GetVideoSourceConfiguration(Media)

- GetVideoSourceConfiguration is supported
- Test Feature: None
- Response: "(VideoSourceConfiguration){\n \_token = \"h264\"\n Name = \"h264\"\n UseCount = 3\n SourceToken = \"h264\"\n Bounds = \n (IntRectangle){\n \_y = 0\n \_width = 1920\n \_x = 0\n \_height = 1080\n }\n }"

### 5.Test: GetVideoEncoderConfigurations(Media)

- GetVideoEncoderConfigurations is supported
- Test Feature: None

- Response: "[(VideoEncoderConfiguration){\n \_token = \"Major\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 2048\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n IPv6Address = \"0.0.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M1.200S\"\n }, (VideoEncoderConfiguration){\n \_token = \"Minor\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 512\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n IPv6Address = \"0.0.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M2.400S\"\n }]"

## 6.Test: GetVideoEncoderConfigurationOptions(Media)

- GetVideoEncoderConfigurationOptions is supported
- Test Feature: None
- Response: "(VideoEncoderConfigurationOptions){\n QualityRange = \n (IntRange){\n Min = 0\n Max = 4\n }\n H264 = \n (H264Options){\n ResolutionsAvailable[] = \n (VideoResolution){\n Width = 1920\n Height = 1080\n },\n GovLengthRange = \n (IntRange){\n Min = 0\n Max = 150\n }\n FrameRateRange = \n (IntRange){\n Min = 1\n Max = 50\n }\n EncodingIntervalRange = \n (IntRange){\n Min = 0\n Max = 0\n }\n }\n Extension = \n (VideoEncoderOptionsExtension){\n H264[] = \n (H264){\n ResolutionsAvailable[] = \n (ResolutionsAvailable){\n Width[] = \n \"1920\",\n Height[] = \n \"1080\",\n },\n GovLengthRange[] = \n (GovLengthRange){\n Min[] = \n \"0\",\n Max[] = \n \"150\",\n },\n FrameRateRange[] = \n (FrameRateRange){\n Min[] = \n \"1\",\n Max[] = \n \"50\",\n },\n EncodingIntervalRange[] = \n (EncodingIntervalRange){\n Min[] = \n \"0\",\n Max[] = \n \"0\",\n },\n BitrateRange[] = \n (BitrateRange){\n Min[] = \n \"0\",\n Max[] = \n \"16384\",\n },\n },\n }\n }"

## 7.Test: GetVideoEncoderConfiguration(Media)

- GetVideoEncoderConfiguration is supported
- Test Feature: None
- Response: "(VideoEncoderConfiguration){\n \_token = \"Major\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 2048\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M1.200S\"\n }"

## 8.Test: GetVideoAnalyticsConfigurations(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 9.Test: GetVideoAnalyticsConfiguration(Media)

- Test Feature: Profile instance has no attribute 'VideoAnalyticsConfiguration'

## 10.Test: GetStreamUri(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 11.Test: GetSnapshotUri(Media)

- GetSnapshotUri is supported
- Test Feature: None

- Response: "(MediaUri){\n Uri = \"rtsp://192.168.15.56:554/stream0/admin/Supervisor\"\n InvalidAfterConnect = False\n InvalidAfterReboot = False\n Timeout = \"P1D\"\n }"

## 12.Test: GetServiceCapabilities(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 13.Test: GetProfiles(Media)

- GetProfiles is supported

- Test Feature: None

- Response: "[(Profile){\n \_token = \"stream0\"\n \_fixed = True\n Name = \"1920\*1080\"\n VideoSourceConfiguration = \n (VideoSourceConfiguration){\n \_token = \"h264\"\n Name = \"h264\"\n UseCount = 3\n SourceToken = \"h264\"\n Bounds = \n (IntRectangle){\n \_y = 0\n \_width = 1920\n \_x = 0\n \_height = 1080\n }\n }\n AudioSourceConfiguration = \n (AudioSourceConfiguration){\n \_token = \"AAC\"\n Name = \"Audio\"\n UseCount = 3\n SourceToken = \"AAC\"\n }\n VideoEncoderConfiguration = \n (VideoEncoderConfiguration){\n \_token = \"Major\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 2048\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M1.200S\"\n }\n AudioEncoderConfiguration = \n (AudioEncoderConfiguration){\n \_token = \"AAC\"\n Name = \"AudioEncoder\"\n UseCount = 3\n Encoding = \"AAC\"\n Bitrate = 64000\n SampleRate = 48000\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0S\"\n }\n }, (Profile){\n \_token = \"stream1\"\n \_fixed = True\n Name = \"704\*576\"\n VideoSourceConfiguration = \n (VideoSourceConfiguration){\n \_token = \"h264\"\n Name = \"h264\"\n UseCount = 3\n SourceToken = \"h264\"\n Bounds = \n (IntRectangle){\n \_y = 0\n \_width = 704\n \_x = 0\n \_height = 576\n }\n }\n AudioSourceConfiguration = \n (AudioSourceConfiguration){\n \_token = \"AAC\"\n Name = \"Audio\"\n UseCount = 3\n SourceToken = \"AAC\"\n }\n VideoEncoderConfiguration = \n (VideoEncoderConfiguration){\n \_token = \"Minor\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 704\n Height = 576\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 512\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M2.400S\"\n }\n AudioEncoderConfiguration = \n (AudioEncoderConfiguration){\n \_token = \"AAC\"\n Name = \"AudioEncoder\"\n UseCount = 3\n Encoding = \"AAC\"\n Bitrate = 64000\n SampleRate = 48000\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0S\"\n }\n }]"

## 14.Test: GetProfile(Media)

- GetProfile is supported

- Test Feature: None

- Response: "(Profile){\n \_token = \"stream0\"\n \_fixed = True\n Name = \"1920\*1080\"\n VideoSourceConfiguration = \n (VideoSourceConfiguration){\n Name = \"h264\"\n UseCount = 3\n SourceToken = None\n Bounds = \n (IntRectangle){\n \_y = 0\n \_width = 1920\n \_x = 0\n \_height = 1080\n }\n }\n AudioSourceConfiguration = \n (AudioSourceConfiguration){\n \_token = \"AAC\"\n Name = \"Audio\"\n UseCount = 3\n SourceToken = \"AAC\"\n }\n VideoEncoderConfiguration = \n (VideoEncoderConfiguration){\n \_token = \"Major\"\n Name = \"h264\"\n UseCount = 1\n Encoding = \"H264\"\n Resolution = \n (VideoResolution){\n Width = 1920\n Height = 1080\n }\n Quality = 0.0\n RateControl = \n (VideoRateControl){\n FrameRateLimit = 30\n EncodingInterval = 0\n BitrateLimit = 2048\n }\n H264 = \n (H264Configuration){\n GovLength = 0\n H264Profile = \"High\"\n }\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0H0M1.200S\"\n }\n AudioEncoderConfiguration = \n (AudioEncoderConfiguration){\n \_token = \"AAC\"\n Name = \"AudioEncoder\"\n UseCount = 1\n Encoding = \"AAC\"\n Bitrate = 64000\n SampleRate = 48000\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n }\n Port = 0\n TTL = 0\n AutoStart = False\n }\n SessionTimeout = \"PT0S\"\n }\n }"

## 15.Test: GetOSDs(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## **16.Test: GetOSDOptions(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **17.Test: GetOSD(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **18.Test: GetMetadataConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **19.Test: GetMetadataConfigurationOptions(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **20.Test: GetMetadataConfiguration(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **21.Test: GetGuaranteedNumberOfVideoEncoderInstances(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **22.Test: GetCompatibleVideoSourceConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **23.Test: GetCompatibleVideoEncoderConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **24.Test: GetCompatibleVideoAnalyticsConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **25.Test: GetCompatibleMetadataConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **26.Test: GetCompatibleAudioSourceConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **27.Test: GetCompatibleAudioOutputConfigurations(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## 28.Test: GetCompatibleAudioDecoderConfigurations(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 29.Test: GetAudioSources(Media)

- GetAudioSources is supported
- Test Feature: None
- Response: "[(AudioSource){\n \_token = \"AAC\"\n Channels = 1\n }]"

## 30.Test: GetAudioSourceConfigurations(Media)

- GetAudioSourceConfigurations is supported
- Test Feature: None
- Response: "[(AudioSourceConfiguration){\n \_token = \"audiosource\_config\_ch01\"\n Name = \"audiosource\_config\_ch01\"\n UseCount = 1\n SourceToken = \"audiosource\_ch01\"\n }]"

## 31.Test: GetAudioSourceConfigurationOptions(Media)

- Test Feature: Type not found: 'ns1:UseCount'

## 32.Test: GetAudioSourceConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 33.Test: GetAudioOutputs(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 34.Test: GetAudioOutputConfigurations(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 35.Test: GetAudioOutputConfigurationOptions(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 36.Test: GetAudioOutputConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## 37.Test: GetAudioEncoderConfigurations(Media)

- GetAudioEncoderConfigurations is supported
- Test Feature: None
- Response: "[(AudioEncoderConfiguration){\n \_token = \"AAC\"\n Name = \"AudioEncoder\"\n UseCount = 1\n Encoding = \"AAC\"\n Bitrate = 64000\n SampleRate = 48000\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"239.168.0.0\"\n } \n Port = 0\n TTL = 0\n AutoStart = False\n } \n SessionTimeout = \"PT0S\" \n }, (AudioEncoderConfiguration){\n \_token = \"AAC\"\n Name = \"AudioEncoder\"\n UseCount = 1\n Encoding = \"AAC\"\n Bitrate = 64000\n SampleRate = 48000\n Multicast = \n (MulticastConfiguration){\n Address = \n (IPAddress){\n

```
Type = \"IPv4\"\\n IPv4Address = \"239.168.0.0\"\\n }\\n Port = 0\\n TTL = 0\\n AutoStart = False\\n }\\n SessionTimeout = \"PT0S\"\\n }]"
```

### 38.Test: GetAudioEncoderConfigurationOptions(Media)

- Test Feature: global name 'config' is not defined

### 39.Test: GetAudioEncoderConfiguration(Media)

- GetAudioEncoderConfiguration is supported
- Test Feature: None
- Response: "(AudioEncoderConfiguration){\\n \_token = \"AAC\"\\n Name = \"AudioEncoder\"\\n UseCount = 1\\n Encoding = \"AAC\"\\n Bitrate = 64000\\n SampleRate = 48000\\n Multicast = \\n (MulticastConfiguration){\\n Address = \\n (IPAddress){\\n Type = \"IPv4\"\\n IPv4Address = \"239.168.0.0\"\\n }\\n Port = 0\\n TTL = 0\\n AutoStart = False\\n }\\n SessionTimeout = \"PT0S\"\\n }"

### 40.Test: GetAudioDecoderConfigurations(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 41.Test: GetAudioDecoderConfigurationOptions(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 42.Test: GetAudioDecoderConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 43.Test: CreateProfile(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 44.Test: AddVideoSourceConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 45.Test: AddVideoEncoderConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 46.Test: AddVideoAnalyticsConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 47.Test: AddPTZConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

### 48.Test: AddMetadataConfiguration(Media)

- Test Feature: Unknown error: (400, u'Bad Request')

## **49.Test: AddAudioSourceConfiguration(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **50.Test: AddAudioOutputConfiguration(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **51.Test: AddAudioEncoderConfiguration(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **52.Test: AddAudioDecoderConfiguration(Media)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **53.Test: SetNetworkDefaultGateway(Device)**

- Test Feature: The DUT did not set new NetworkDefaultGateway
- Response: "None"

## **54.Test: SetHostname(Device)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **55.Test: SetDiscoveryMode(Device)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **56.Test: RemoveScopes(Device)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **57.Test: GetUsers(Device)**

- GetUsers is supported
- Test Feature: None
- Response: "[{User}]\n Username = \"admin\"\n Password = \"Supervisor\"\n UserLevel = \"Administrator\"\n }]"

## **58.Test: GetSystemUris(Device)**

- Test Feature: Unknown error: (400, u'Bad Request')

## **59.Test: GetSystemDateAndTime(Device)**

- GetSystemDateAndTime is supported
- Test Feature: None

- Response: "(SystemDateTime){\n DateTimeType = \"Manual\"\n DaylightSavings = False\n TimeZone = \n (TimeZone){\n TZ = \"CST-8\"\n }\n UTCDateTime = \n (DateTime){\n Time = \n (Time){\n Hour = 21\n Minute = 57\n Second = 19\n }\n Date = \n (Date){\n Year = 2019\n Month = 4\n Day = 17\n }\n }\n LocalDateTime = \n (DateTime){\n Time = \n (Time){\n Hour = 0\n Minute = 57\n Second = 19\n }\n Date = \n (Date){\n Year = 2019\n Month = 4\n Day = 18\n }\n }\n }"

## 60.Test: GetSupportedServices(Device)

- GetSupportedServices is supported
- Test Feature: None
- Response: [ { "name": "Devicemgmt", "supported": true }, { "name": "Media", "supported": true }, { "name": "Imaging", "supported": false }, { "name": "Analytics", "supported": false }, { "name": "PTZ", "supported": false }, { "name": "DeviceIO", "supported": false }, { "name": "Events", "supported": false }, { "name": "Replay", "supported": false }, { "name": "Recording", "supported": false }, { "name": "Search", "supported": false }, { "name": "Pullpoint", "supported": false }, { "name": "Receiver", "supported": false } ]

## 61.Test: GetServices(Device)

- GetServices is supported
- Test Feature: The DUT send a valid response in both cases(IncludeCapability)
- Response: "[(Service){\n Namespace = \"http://www.onvif.org/ver10/device/wsd\"\n XAddr = \"http://192.168.15.56:8080/onvif/device\_service\"\n Version = \n (OnvifVersion){\n Major = 1\n Minor = 40\n }\n }, (Service){\n Namespace = \"http://www.onvif.org/ver10/media/wsd\"\n XAddr = \"http://192.168.15.56:8080/onvif/device\_service\"\n Version = \n (OnvifVersion){\n Major = 1\n Minor = 50\n }\n }, (Service){\n Namespace = \"http://www.onvif.org/ver10/network/wsd\"\n XAddr = \"http://192.168.15.56:8080/onvif/device\_service\"\n Version = \n (OnvifVersion){\n Major = 1\n Minor = 0\n }\n }]"

## 62.Test: GetScopes(Device)

- GetScopes is supported
- Test Feature: None
- Response: "[(Scope){\n ScopeDef = \"Fixed\"\n ScopeItem = \"onvif://www.onvif.org/type/video\_encoder\"\n }, (Scope){\n ScopeDef = \"Fixed\"\n ScopeItem = \"onvif://www.onvif.org/type/audio\_encoder\"\n }, (Scope){\n ScopeDef = \"Fixed\"\n ScopeItem = \"onvif://www.onvif.org/hardware/encoder\_model\"\n }, (Scope){\n ScopeDef = \"Fixed\"\n ScopeItem = \"onvif://www.onvif.org/name/codec\_encoder\"\n }, (Scope){\n ScopeDef = \"Fixed\"\n ScopeItem = \"onvif://www.onvif.org/location/office\"\n }]"

## 63.Test: GetNetworkProtocols(Device)

- GetNetworkProtocols is supported
- Test Feature: None
- Response: "[(NetworkProtocol){\n Name = \"HTTP\"\n Enabled = True\n Port[] = \n 80,\n }, (NetworkProtocol){\n Name = \"RTSP\"\n Enabled = True\n Port[] = \n 554,\n }]"

## 64.Test: GetNetworkInterfaces(Device)

- GetNetworkInterfaces is supported
- Test Feature: None
- Response: "[(NetworkInterface){\n \_token = \"eth0\"\n Enabled = True\n Info = \n (NetworkInterfaceInfo){\n Name = \"eth0\"\n HwAddress = \"4c:64:d9:f0:13:cc\"\n }\n Link = \n (NetworkInterfaceLink){\n AdminSettings = \n (NetworkInterfaceConnectionSetting){\n AutoNegotiation = True\n Speed = 100\n Duplex = \"Full\"\n }\n OperSettings = \n (NetworkInterfaceConnectionSetting){\n AutoNegotiation = True\n Speed = 100\n Duplex = \"Full\"\n }\n InterfaceType =



```
"0"\n }\n IPv4 = \n (IPv4NetworkInterface){\n Enabled = True\n Config = \n (IPv4Configuration){\n Manual[] = \n (PrefixedIPv4Address){\n Address = \"192.168.15.56\"\n PrefixLength = 24\n },\n FromDHCP = \n (PrefixedIPv4Address){\n Address = \"192.168.15.56\"\n PrefixLength = 24\n }\n DHCP = True\n }\n }\n }"
```

## 65.Test: GetNetworkDefaultGateway(Device)

- GetNetworkDefaultGateway is supported
- Test Feature: None
- Response: "(NetworkGateway){\n IPv4Address[] = \n \"192.168.15.1\", \n }"

## 66.Test: GetNTP(Device)

- GetNTP is supported
- Test Feature: None
- Response: "(NTPInformation){\n FromDHCP = False\n NTPManual[] = \n (NetworkHost){\n Type = \"DNS\"\n DNSName = \"pool.ntp.org\"\n }, \n }"

## 67.Test: GetHostname(Device)

- GetHostname is supported
- Test Feature: None
- Response: "(HostnameInformation){\n FromDHCP = False\n Name = \"Codec-120\"\n }"

## 68.Test: GetDiscoveryMode(Device)

- GetDiscoveryMode is supported
- Test Feature: This operation got the discovery mode of a device
- Response: "Discoverable"

## 69.Test: GetDeviceInformation(Device)

- GetDeviceInformation is supported
- Test Feature: None
- Response: "(reply){\n Manufacturer = \"codec\"\n Model = \"Encoder\"\n FirmwareVersion = \"V1.76\"\n SerialNumber = \"CM150311806040141\"\n HardwareId = \"0.2\"\n }"

## 70.Test: GetDNS(Device)

- GetDNS is supported
- Test Feature: None
- Response: "(DNSInformation){\n FromDHCP = False\n DNSManual[] = \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"192.168.0.210\"\n }, \n (IPAddress){\n Type = \"IPv4\"\n IPv4Address = \"8.8.8.8\"\n }, \n }"

## 71.Test: GetCapabilities(Device)

- GetCapabilities is supported
- Test Feature: None

- Response: "(Capabilities){\n Analytics = \n (AnalyticsCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n RuleSupport = False\n AnalyticsModuleSupport = False\n }\n Device = \n (DeviceCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n Network = \n (NetworkCapabilities){\n IPFilter = True\n ZeroConfiguration = False\n IPVersion6 = False\n DynDNS = True\n }\n System = \n (SystemCapabilities){\n DiscoveryResolve = False\n DiscoveryBye = False\n RemoteDiscovery = False\n SystemBackup = False\n SystemLogging = False\n FirmwareUpgrade = False\n SupportedVersions[] = \n (OnvifVersion){\n Major = 1\n Minor = 0\n },\n }\n IO = \n (IOCapabilities){\n InputConnectors = 0\n RelayOutputs = 0\n }\n Security = \n (SecurityCapabilities){\n TLS1.1 = False\n TLS1.2 = False\n OnboardKeyGeneration = False\n AccessPolicyConfig = False\n X.509Token = False\n SAMLToken = False\n KerberosToken = False\n RELToken = False\n }\n }\n Events = \n (EventCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n WSSubscriptionPolicySupport = False\n WSPullPointSupport = False\n WSPausableSubscriptionManagerInterfaceSupport = False\n }\n Imaging = \n (ImagingCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n }\n Media = \n (MediaCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n StreamingCapabilities = \n (RealTimeStreamingCapabilities){\n RTPMulticast = True\n RTP\_TCP = False\n RTP\_RTSP\_TCP = True\n Extension = \"\"\n }\n Extension[] = \n (Extension){\n ProfileCapabilities[] = \n (ProfileCapabilities){\n MaximumNumberOfProfiles[] = \n \"3\",,\n },,\n }\n }\n PTZ = \n (PTZCapabilities){\n XAddr = \"http://192.168.15.56:8080/onvif\"\n }\n }"

## 72.Test: CreateUsers(Device)

- CreateUsers is supported
- Test Feature: The DUT created an user with Username: lalalal
- Response: "None"

## 73.Test: AddScopes(Device)

- Test Feature: Unknown error: (400, u'Bad Request')