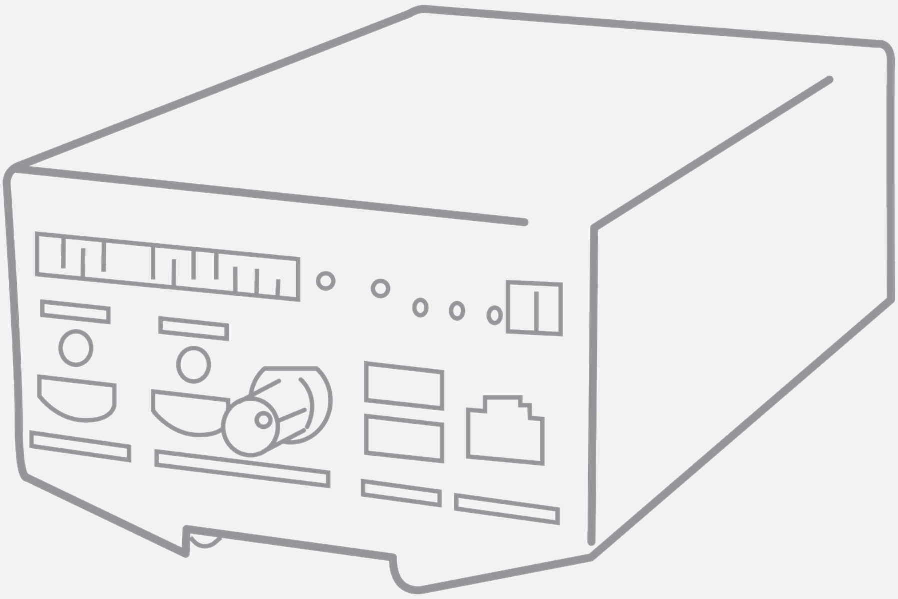
USER GUIDE

Picolo.net HD1



This documentation is provided with Picolo.net HD1 2.7.3 (doc build 3038).
www.euresys.com

Contents

PART I

USING THE WEB INTERFACE

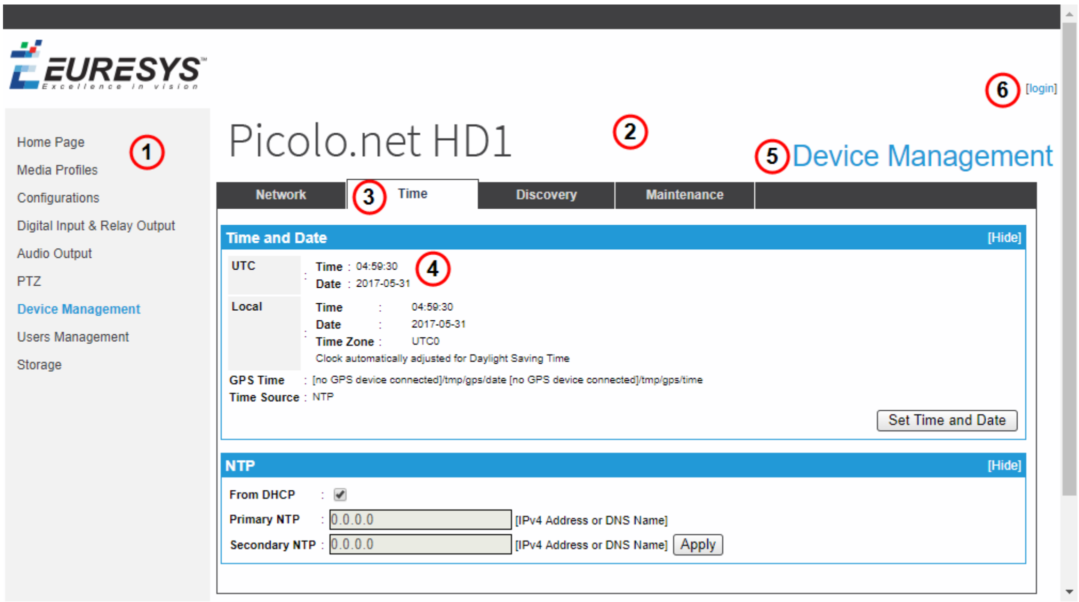
1. Introduction

URL

The Home Page URL is: `http://[device-ip-address]`

The web pages of Picolo.net HD1 are available in English (default), Japanese, Chinese and Korean. The selection is automatic based on the 'Accept-Language' HTTP header sent by your web browser (it usually depends on your operating system localization).

Page Layout



Time and Date [Hide]

UTC : Time : 04:59:30
Date : 2017-05-31

Local : Time : 04:59:30
Date : 2017-05-31
Time Zone : UTC0
Clock automatically adjusted for Daylight Saving Time

GPS Time : [no GPS device connected]/tmp/gps/date [no GPS device connected]/tmp/gps/time
Time Source : NTP

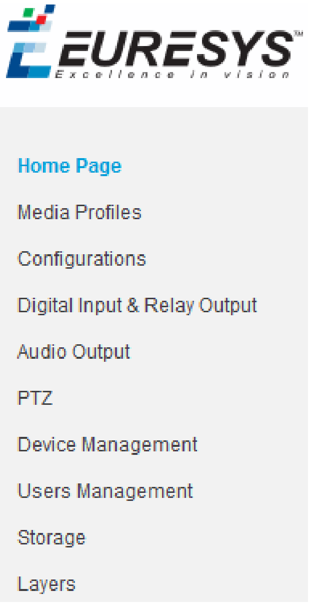
NTP [Hide]

From DHCP : ☑
Primary NTP : 0.0.0.0 [IPv4 Address or DNS Name]
Secondary NTP : 0.0.0.0 [IPv4 Address or DNS Name]

Page sample

1. Left panel: Navigation links
2. Main panel
3. Tab
4. Panel
5. Page title
6. Login

Navigation Links



Home Page

Media Profiles

Configurations

Digital Input & Relay Output

Audio Output

PTZ

Device Management

Users Management

Storage

Layers

Navigation Links

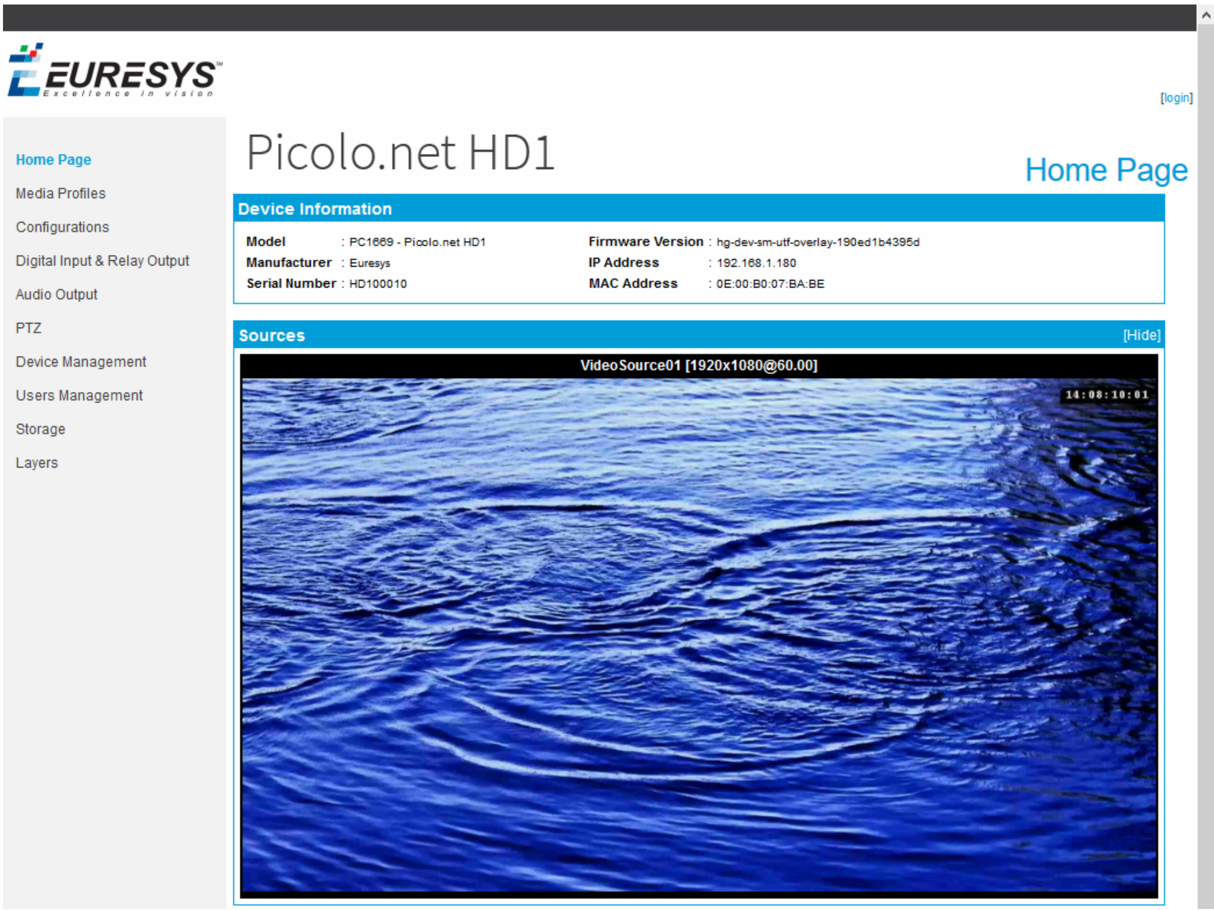
Navigation links provide a single-click access to the main page of each section.

Select:

- "Home Page" on page 8 to view device information and display video source.
- "Media Profiles Page" on page 13 to view/edit/delete/create media profiles.
- "Configurations Page" on page 19 to view/edit configurations of video source, video encoder, audio source, audio encoder, PTZ and metadata objects.
- "Digital Inputs & Relay Outputs Page" on page 34 to view/edit configuration of digital input and relay output objects.
- "Audio Outputs Page" on page 36 to view/edit configuration of audio output object
- "PTZ Page" on page 37 to view/edit configuration of the serial port and the PTZNode objects
- "Device Management Page" on page 41 to view/edit network, time and date and discovery settings and perform maintenance tasks
- "Users Management Page" on page 49 to create/delete users and view/edit user properties.
- "Storage Page" on page 52 to mount/unmount storage media, enable/disable/start/stop recording and list/preview stored media files.
- "Layers Page" on page 55 to configure OSD (On Screen Display) location and content.

2. Home Page

View device information and display video source



Picolo.net HD1

Home Page

Device Information

Model	: PC1669 - Picolo.net HD1	Firmware Version	: hg-dev-sm-utf-overlay-190ed1b4395d
Manufacturer	: Euresys	IP Address	: 192.168.1.180
Serial Number	: HD100010	MAC Address	: 0E:00:B0:07:BA:BE

Sources

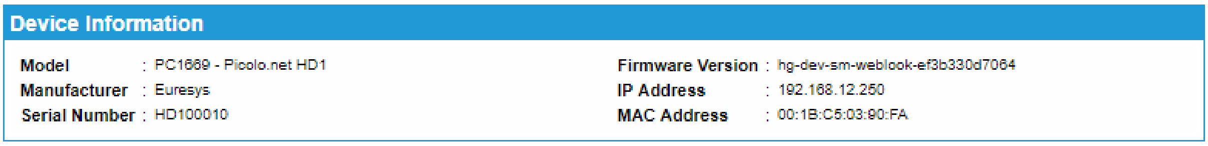
VideoSource01 [1920x1080@60.00]

Home page

The main pane of the Home page displays 2 panels:

- A Device Information panel providing general information about the device
- A Sources panel providing a mosaic display of all the video sources of the device

Device Information Panel



Device Information

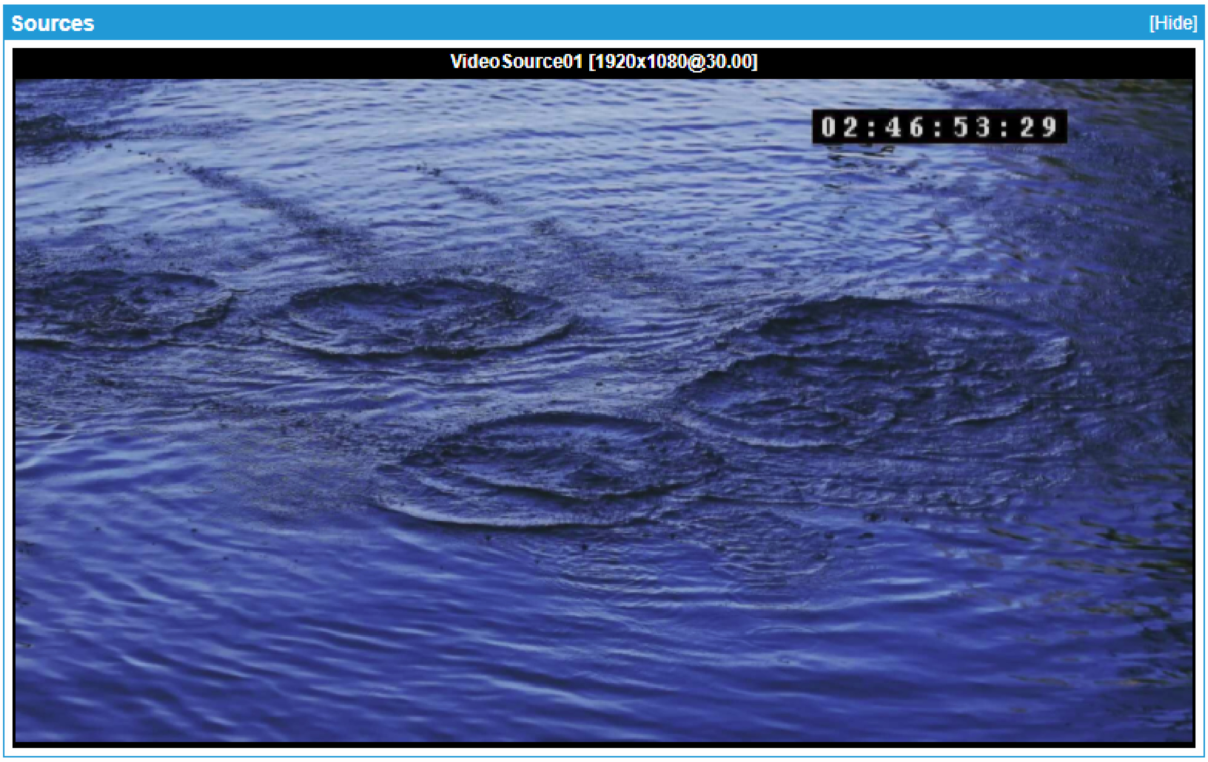
Model	:	PC1669 - Picolo.net HD1	Firmware Version	:	hg-dev-sm-weblook-ef3b330d7064
Manufacturer	:	Euresys	IP Address	:	192.168.12.250
Serial Number	:	HD100010	MAC Address	:	00:1B:C5:03:90:FA

Device Information panel

Device Information panel fields description

Name	Description
Model	Product code and product name of the device
Manufacturer	Manufacturer name of the device
Serial Number	Serial number of the device
Firmware Version	Major and minor version numbers of the firmware that is currently on the device.
IP Address	IPv4 address of the device currently assigned to the device
MAC Address	MAC Address of the LAN port of the device

Sources Panel



Sources panel

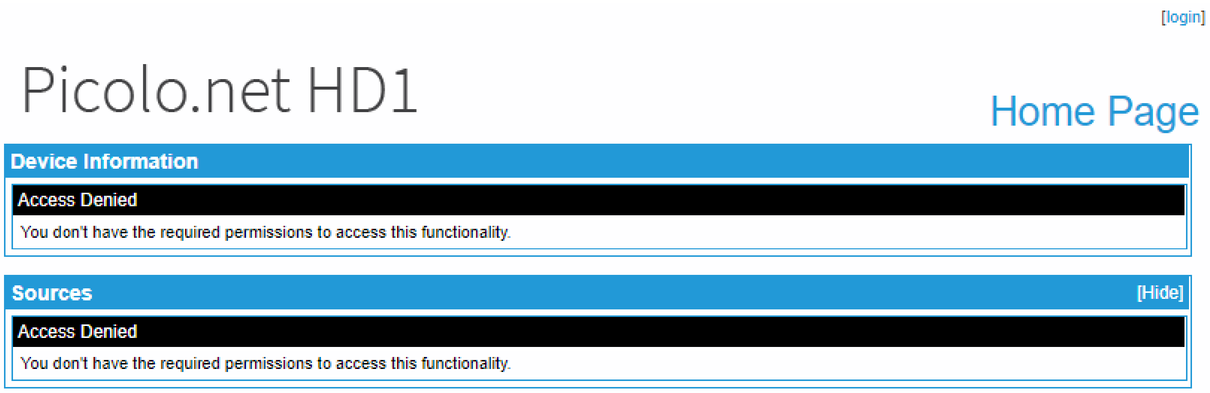
The Sources panel shows a rectangular area containing:

- A title composed of the name, the native resolution, and the native frame rate of the video source.
- A snapshot image providing that the source is referenced by a properly configured ONVIF Media Profile.

If the ONVIF Media Profile is not properly configured, the image is replaced by a black background overlayed by a crossed rectangle.
If the source has no video, a blue image is displayed.
Clicking on the image brings the browser to the View/Edit Profile page for the profile that generated the snapshot.

Access Denied Home Page

Once security is enabled, an anonymous user accessing the device Home page obtains the following page:



[login]

Picolo.net HD1

Home Page

Device Information

Access Denied

You don't have the required permissions to access this functionality.

Sources [Hide]

Access Denied

You don't have the required permissions to access this functionality.

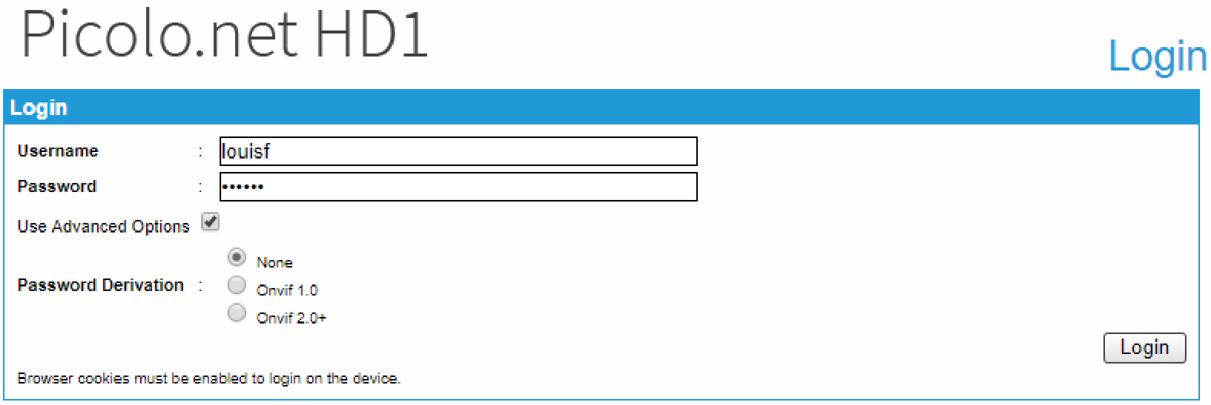
Home page when access is denied

Clicking on the [login] hyperlink opens the Login page.

2.1. Login Page

The Login page displays the Login panel.

Login panel



Picolo.net HD1

Login

Login

Username : louisf

Password : ••••••

Use Advanced Options ☑

Password Derivation : ● None ○ Onvif 1.0 ○ Onvif 2.0+

Browser cookies must be enabled to login on the device.

Login panel

Login panel fields description

Name	Description
Username	User name
Password	User password
Use Advanced Options	Cross the checkbox if specific password derivations are required.

Password Derivation

Password derivation allows the user of multiple devices to type the same string when authenticating on any device while the value stored on the device is actually different for each device.

Value	Description
None	No password derivation, the device password is directly typed by the user. Default setting.
Onvif 1.0	The password is computed (derived) from the device identity and the user-typed string according to ONVIF 1.0 specification.
Onvif 2.0+	The password is computed (derived) from the device identity and the user-typed string according to ONVIF 2.0 (or later) specification.

Users account lockout after repeated login failures

To better protect your device and your private data against brute-force attacks, every user account has:

- A failure counter.
- A time stamp of the oldest remembered failure.

If the maximum number of attempts is reached, any additional login attempt is ignored with a "423 locked user" error message until the timer expires.

TIP
You can configure the maximum number of attempts and the timer duration in the "Users Management Page" on page 49.

The access log reports:

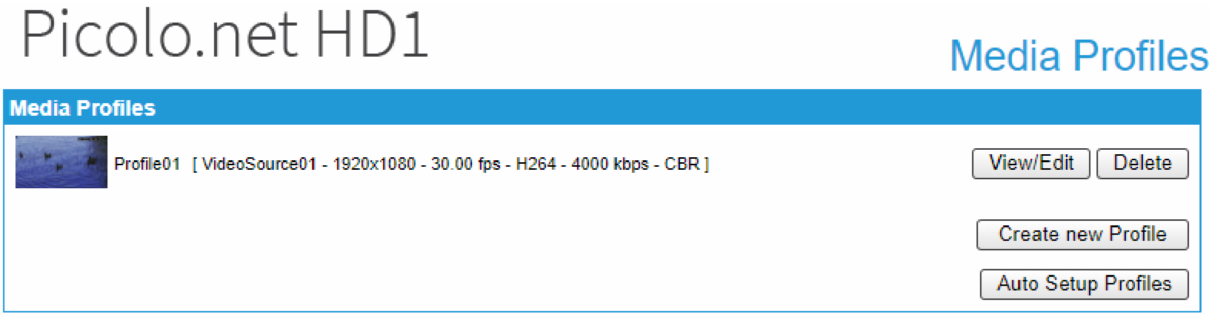
- `authentication failure` when the entered password does not match the one stored on the device.
- `user account locked` when the password comparison is skipped due to the auto-lockout protection.

An administrator can unlock a blocked account at any time from the users management web page.

3. Media Profiles Page

The Media Profiles page displays the Media Profiles panel.

Media Profiles panel


Picolo.net HD1

Media Profiles

Media Profiles

Profile01 [VideoSource01 - 1920x1080 - 30.00 fps - H264 - 4000 kbps - CBR]

Media Profiles panel

Profiles List

The Media Profiles panel lists all the existing ONVIF Media Profiles.
Each list item contains:

- A thumbnail image of the video source
- The name of the profile e.g. `Profile01`
- Between square brackets, a selection of profile properties including: name of the video source, resolution, frame rate, encoding method, bit rate, and rate control method of the encoded stream.
- A View/Edit button.
- A Delete button.

Clicking on the View/Edit button opens the Media Profile page allowing the user to view or edit the profile properties.

Clicking on the Delete button deletes the profile.

Profile Creation & Auto Setup

The bottom right area of the Media Profiles panel contains two buttons:

- The Create New Profile button.
- The Auto Setup Profiles button.

Clicking on the Create New Profile button starts the profile creation procedure. First of all, the procedure opens a dialog box requiring the name of the new profile. Then it displays the Configurations page allowing the user to configure the ONVIF Media Profile.

Clicking on the Auto Setup Profiles button initiates the auto setup procedure. Before proceeding, a dialog box opens requiring to confirm the action.

> **WARNING**
> The auto setup procedure erases all the existing ONVIF Media Profiles.

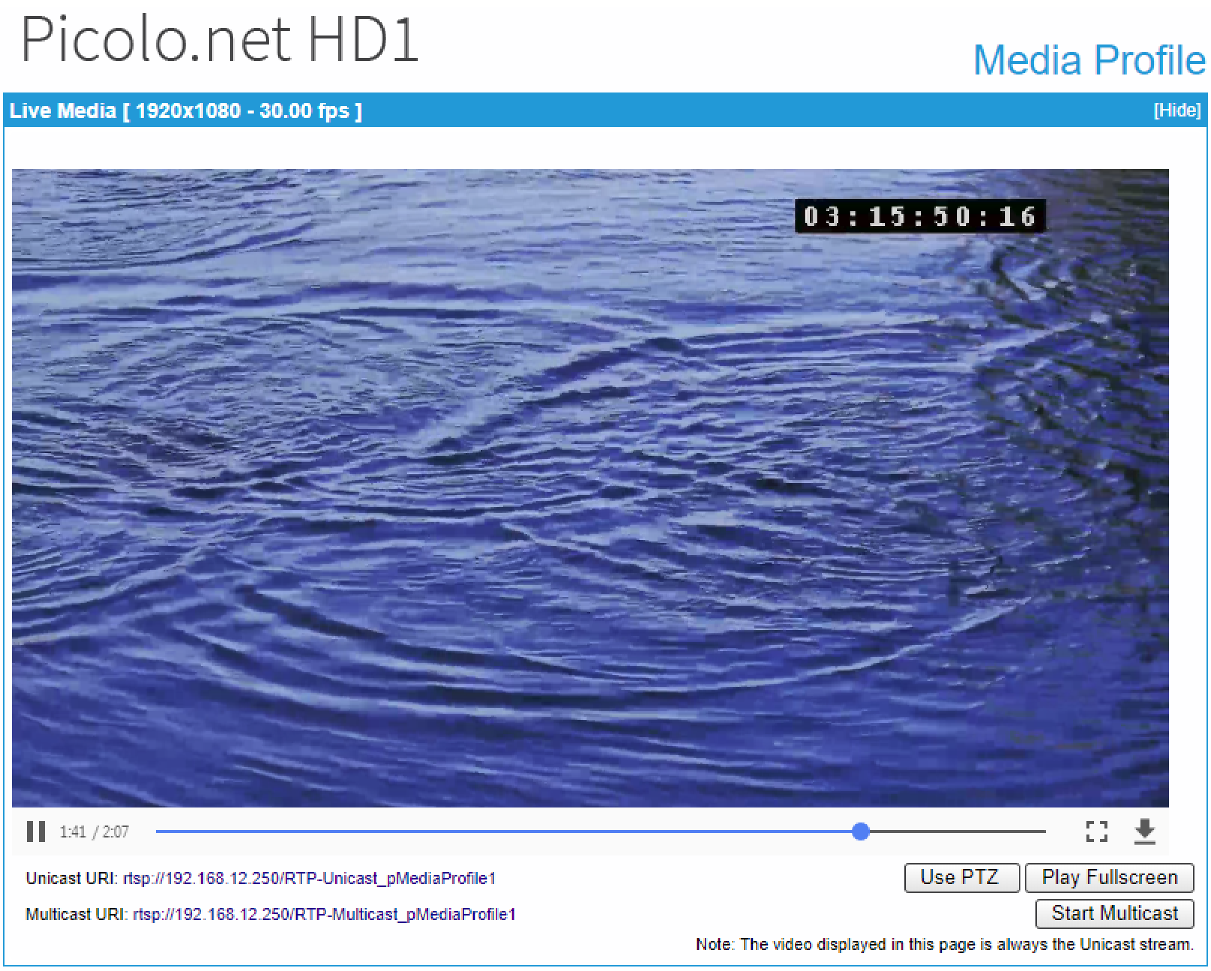
3.1. Media Profile Page

The Media Profile page of the Web Server is relative to a single ONVIF Media Profile. It allows the user to:

- View the encoded video stream in the Live Media panel
- View the properties of the components of an ONVIF Media Profile using the configuration panels
- Modify the composition of ONVIF Media Profiles using the Media Profile Configuration panels

The panels composing this page can be hidden or shown individually by clicking on the [Hide] or [Show] text. Initially, only the Live Media panel is shown.

Live Media Panel



Unicast URI: rtsp://192.168.12.250/RTP-Unicast_pMediaProfile1

Multicast URI: rtsp://192.168.12.250/RTP-Multicast_pMediaProfile1

Note: The video displayed in this page is always the Unicast stream.

Live Media panel

The Live Media panel provides a live display of the video source unicast stream using the VLC plug-in of the Web Browser.

The panel title shows, between square brackets, the resolution and the frame rate of the encoded video stream.

In the bottom area, the panel provides:

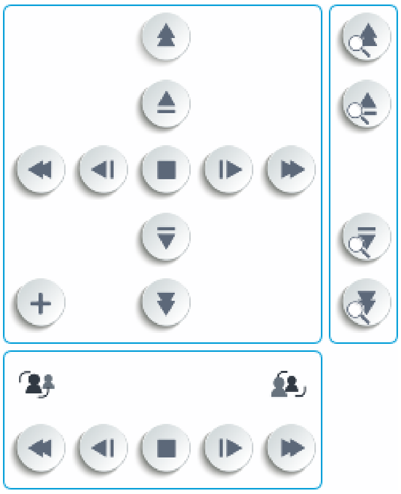
- The Unicast URL of the video stream
- The Multicast URL of the video stream

The Start Multicast button starts multicast streaming for the selected media profile.

- This is not necessary for clients that connect to the stream via the RTSP link provided.
- Multicast streaming continues until explicitly stopped (even after a reboot of the device).
- This button also starts RFC2974 session announcement messages for the corresponding media profile.

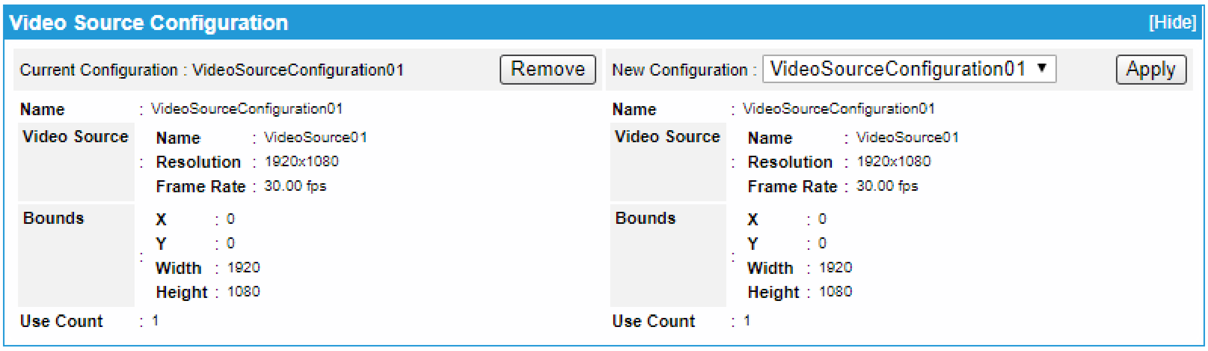
The Play Fullscreen button enlarges the live video on the entire screen.

The Use PTZ button adds PTZ controls on the right side of the image as shown on the following image:



PTZ controls

Media Profile Configuration Panels

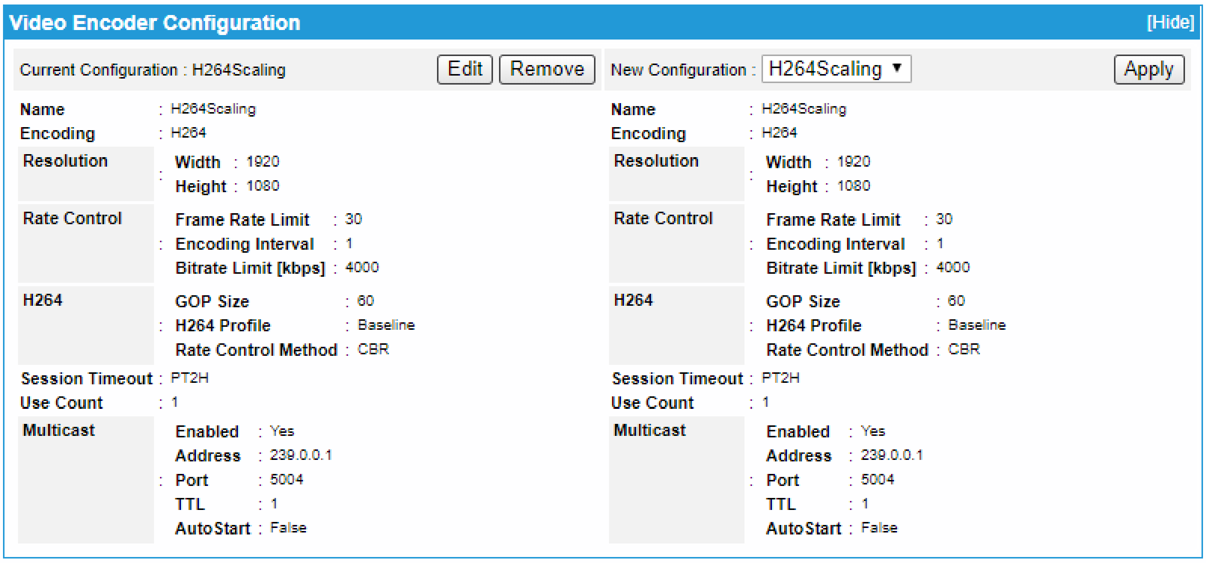


Video Source Configuration [Hide]

Current Configuration : VideoSourceConfiguration01 Remove | New Configuration : VideoSourceConfiguration01 ▼ Apply

Name : VideoSourceConfiguration01	**Name** : VideoSourceConfiguration01
Video Source : **Name** : VideoSource01 **Resolution** : 1920x1080 **Frame Rate** : 30.00 fps	**Video Source** : **Name** : VideoSource01 **Resolution** : 1920x1080 **Frame Rate** : 30.00 fps
Bounds : **X** : 0 **Y** : 0 **Width** : 1920 **Height** : 1080	**Bounds** : **X** : 0 **Y** : 0 **Width** : 1920 **Height** : 1080
Use Count : 1	**Use Count** : 1

Video Source Configuration panel

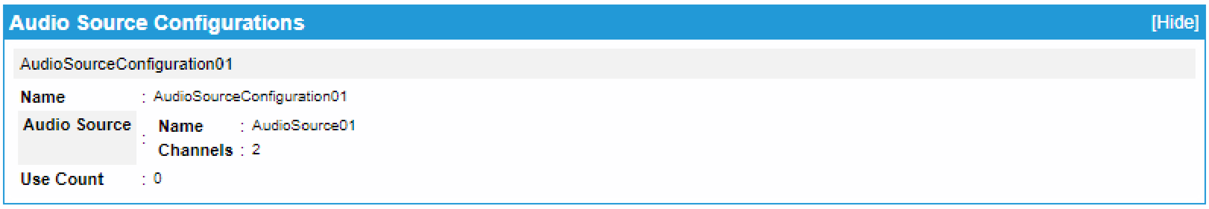


Video Encoder Configuration [Hide]

Current Configuration : H264Scaling Edit Remove | New Configuration : H264Scaling ▼ Apply

Name : H264Scaling	**Name** : H264Scaling
Encoding : H264	**Encoding** : H264
Resolution : **Width** : 1920 **Height** : 1080	**Resolution** : **Width** : 1920 **Height** : 1080
Rate Control : **Frame Rate Limit** : 30 **Encoding Interval** : 1 **Bitrate Limit [kbps]** : 4000	**Rate Control** : **Frame Rate Limit** : 30 **Encoding Interval** : 1 **Bitrate Limit [kbps]** : 4000
H264 : **GOP Size** : 60 **H264 Profile** : Baseline **Rate Control Method** : CBR	**H264** : **GOP Size** : 60 **H264 Profile** : Baseline **Rate Control Method** : CBR
Session Timeout : PT2H	**Session Timeout** : PT2H
Use Count : 1	**Use Count** : 1
Multicast : **Enabled** : Yes **Address** : 239.0.0.1 **Port** : 5004 **TTL** : 1 **AutoStart** : False	**Multicast** : **Enabled** : Yes **Address** : 239.0.0.1 **Port** : 5004 **TTL** : 1 **AutoStart** : False

Video Encoder Configuration panel



Audio Source Configurations [Hide]

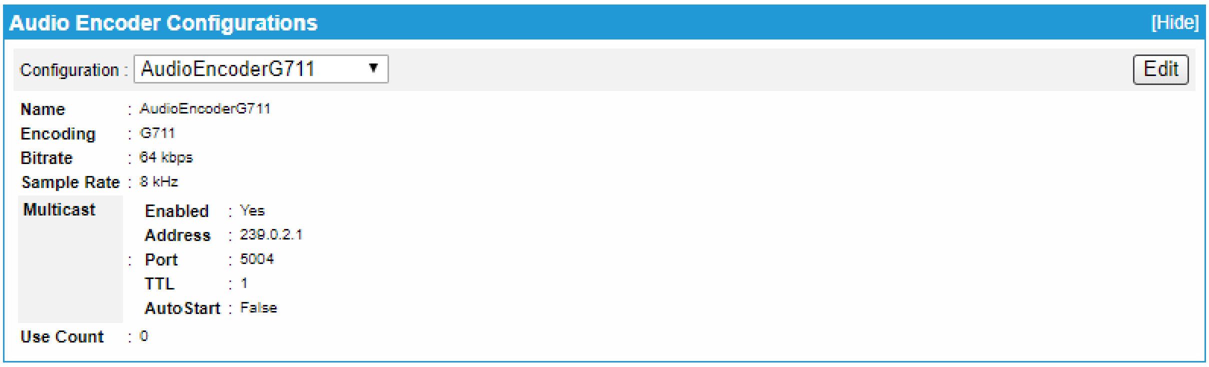
AudioSourceConfiguration01

Name : AudioSourceConfiguration01

Audio Source : **Name** : AudioSource01
Channels : 2

Use Count : 0

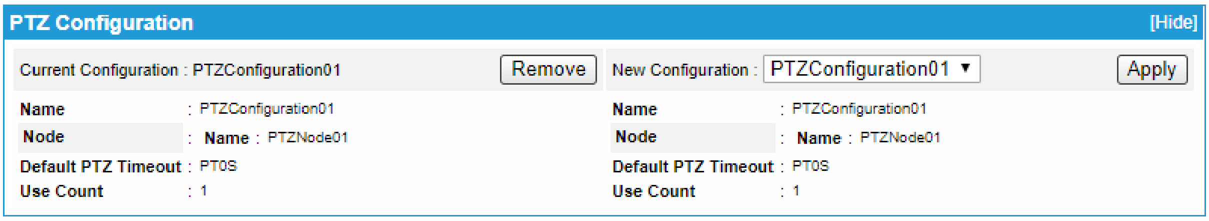
Audio Source Configurations panel



Configuration : AudioEncoderG711

Name : AudioEncoderG711
Encoding : G711
Bitrate : 64 kbps
Sample Rate : 8 kHz
Multicast : Enabled : Yes
Address : 239.0.2.1
Port : 5004
TTL : 1
AutoStart : False
Use Count : 0

Audio Encoder Configurations panel



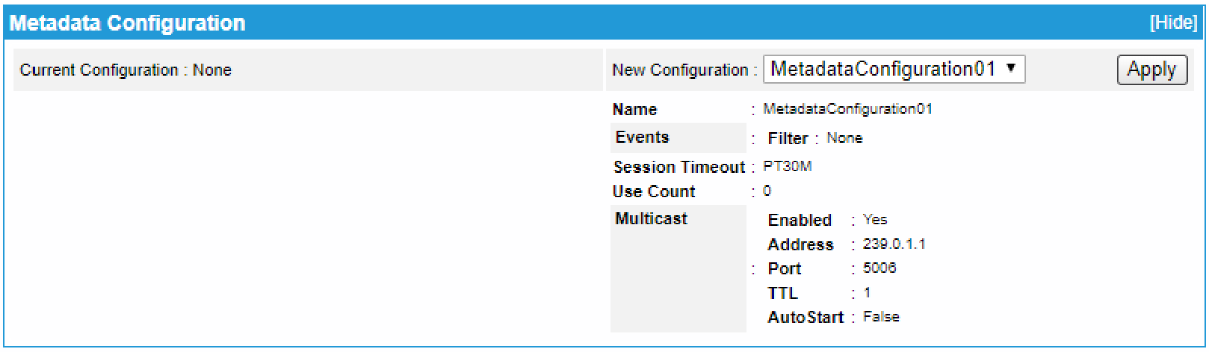
Current Configuration : PTZConfiguration01

Name : PTZConfiguration01
Node : Name : PTZNode01
Default PTZ Timeout : PT0S
Use Count : 1

New Configuration : PTZConfiguration01

Name : PTZConfiguration01
Node : Name : PTZNode01
Default PTZ Timeout : PT0S
Use Count : 1

PTZ Configuration panel



Current Configuration : None

New Configuration : MetadataConfiguration01

Name : MetadataConfiguration01
Events : Filter : None
Session Timeout : PT30M
Use Count : 0
Multicast : Enabled : Yes
Address : 239.0.1.1
Port : 5006
TTL : 1
AutoStart : False

Metadata Configuration panel

- The configuration panels of the Media Profile page allow to:
  - View the composition of the profile and the characteristics their components
  - Modify the composition of the profile by addition or deletion of components.
- To facilitate the modification of existing ONVIF Media Profiles, each panel shows simultaneously for each component:
  - On the left side: the characteristics of the configuration that is currently used by the ONVIF Media Profile
  - On the right side: the characteristics of any selectable configuration
- Providing that the component is currently used in the profile, the upper left quadrant shows :

- □ The name of the current configuration
- □ A Remove button
- □ An Edit button (only on relevant panels)

- Clicking on the Remove button removes the component from the profile.
- Clicking on the Edit button opens the Edit Configuration panel of the component allowing the user to edit its properties.
- The upper right quadrant shows:
  - □ A drop-down box allowing the user to select a new configuration.
  - □ An Apply button.
- Clicking on the Apply button applies the new configuration to the profile.

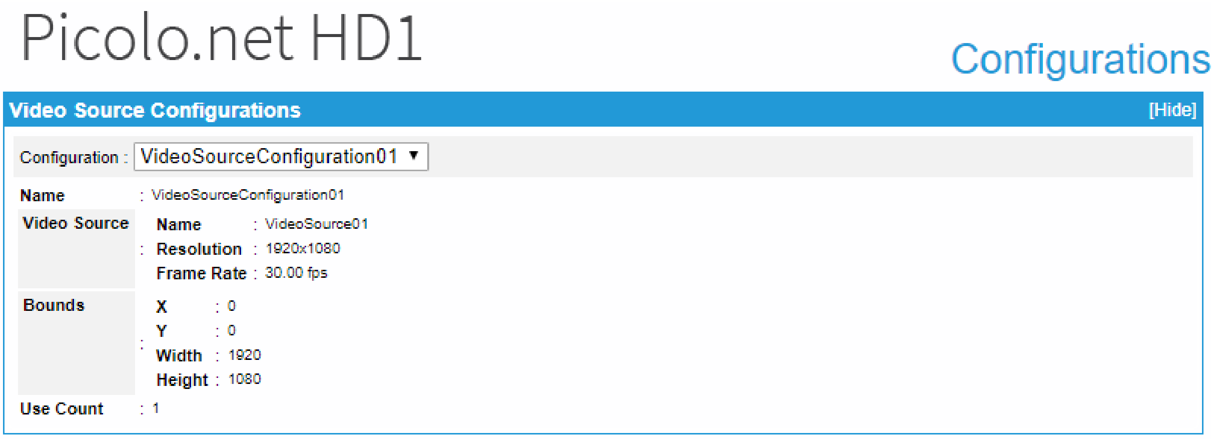
4. Configurations Page

The Configurations page allows the user to view or edit the configurations of all the software objects. It provides six panels, one for each component type of an ONVIF Media Profile:

- "Video Source Configurations panel" on page 20
- "Video Encoder Configurations panels" on page 21
- "Audio Source Configurations panel" on page 23
- "Audio Encoder Configurations panel" on page 24
- "PTZ Configurations panel" on page 25
- "Metadata Configurations panel" on page 26

All panels composing this pane can be hidden or shown individually by clicking on the [Hide] or [Show] text.

Video Source Configurations panel



Picolo.net HD1

Configurations

Video Source Configurations [Hide]

Configuration : VideoSourceConfiguration01

Name : VideoSourceConfiguration01

Video Source : Name : VideoSource01 Resolution : 1920x1080 Frame Rate : 30.00 fps

Bounds : X : 0 Y : 0 Width : 1920 Height : 1080

Use Count : 1

Video Source Configurations panel

- Select a `VideoSourceConfiguration` object from the Configuration drop-down list in the upper area.

Video source field description

Name	Description
Name	The name of the `VideoSourceConfiguration` object
Video Source - Name	The name of the video source
Video Source - Resolution	The resolution [H x V] of the video source, e.g. 1920x1080
Video Source - Frame Rate	The frame rate of the video source, expressed in fps, e.g.30.00 fps
Bounds - X, Y	The position offset of the acquired image relative to the camera active area
Bounds - Width	The number of columns of the acquired image
Bounds - Height	The number of lines of the acquired image
Use Count	The number of ONVIF Media Profiles using this object

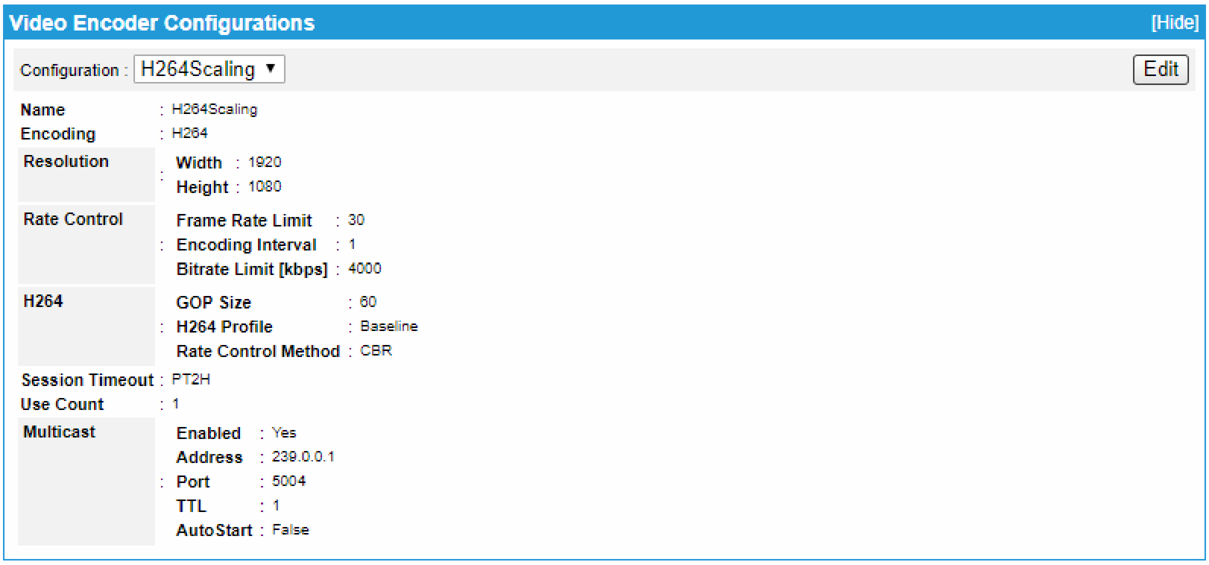
Video Encoder Configurations panels

The layout of the Video Encoder Configurations panel is specific to the video encoding method: H.264/ H.265 or JPEG.

View and edit the video encoder configuration

- Select a video encoding method from the Configuration drop-down list in the upper area.
- Click Edit in the upper area of the panel to open the Video Encoder Configuration Edition page.

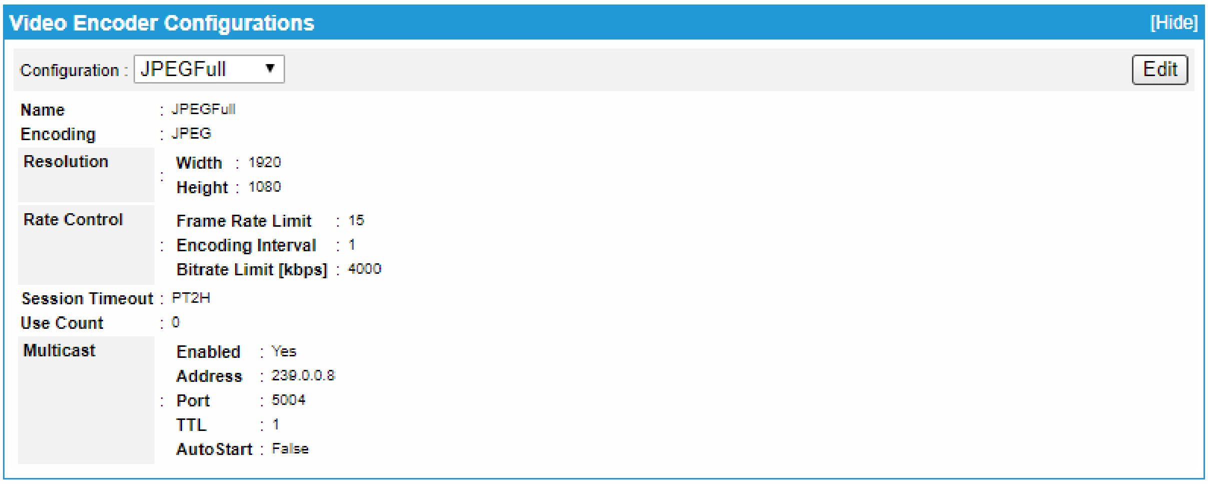
The lower area of the panel shows the properties related to the selected encoding method.



Configuration : H264Scaling ▼

Name : H264Scaling
Encoding : H264
Resolution : Width : 1920
Height : 1080
Rate Control : Frame Rate Limit : 30
Encoding Interval : 1
Bitrate Limit [kbps] : 4000
H264 : GOP Size : 60
H264 Profile : Baseline
Rate Control Method : CBR
Session Timeout : PT2H
Use Count : 1
Multicast : Enabled : Yes
Address : 239.0.0.1
Port : 5004
TTL : 1
AutoStart : False

Video Encoder Configurations panel - H.264/H.265 case



Configuration : JPEGFull ▼

Name : JPEGFull
Encoding : JPEG
Resolution : Width : 1920
Height : 1080
Rate Control : Frame Rate Limit : 15
Encoding Interval : 1
Bitrate Limit [kbps] : 4000
Session Timeout : PT2H
Use Count : 0
Multicast : Enabled : Yes
Address : 239.0.0.8
Port : 5004
TTL : 1
AutoStart : False

Video Encoder Configurations panel - JPEG case

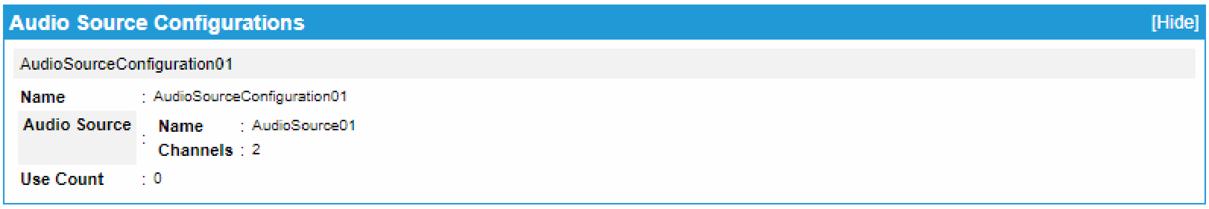
Video encoding field description

A cross on one or both columns on the right of the table means specify whether the field is available as a parameter for the given video encoding method:

Name	**Description**	**H.264** **H.265**	**JPEG**
Name	Token name of the `VideoEncoderConfiguration` object, e.g. `VideoEncoderConfiguration01`	X	X
Encoding	Used video codec.	X	X
Resolution			
Width Height	Image size of the encoded stream	X	X
Rate Control			
Frame Rate Limit	Maximum output frame rate in fps.	X	X
Encoding Interval	Interval at which images are encoded and transmitted.	X	X
Bitrate Limit	Maximum output bit rate in kbps	X	X
Session Timeout	RTSP session timeout. *The duration is expressed using the W3C lexical representation: PnYn MnDTnH nMnS*	X	X
Use Count	Number of ONVIF Media Profiles using that Video Encoder Configuration.	X	X
Multicast			
Enabled	Indicates if the RTP multicast streaming of the encoded video is properly configured with a non-zero IP address and port number.	X	X
Address	IP address of the multicast group. *In IPv4, addresses 224.0.0.0 through 239.255.255.255 are designated as multicast addresses.*	X	X
Port	Port number of the multicast group.	X	X
TTL	Time-To-Live of the multicast IP datagrams. *Usually 1 since the datagrams stops after the first router.*	X	X
AutoStart	Indicates the persistence of multicast streaming. *When true, the multicast streaming starts automatically.*	X	X
H264		X	
GOP Size	Group of Pictures (or Video frames) length.	X	

Name	Description	H.264 H.265	JPEG
H264 Profile	H.264 profile: baseline, main or high.	X	
Rate Control Method	Rate control method of the H.264 method: ● VBR: Variable Bit Rate ● CBR: Constant Bit Rate	X	

Audio Source Configurations panel



AudioSourceConfiguration01

Name : AudioSourceConfiguration01

Audio Source : Name : AudioSource01

Channels : 2

Use Count : 0

Audio Source Configurations panel

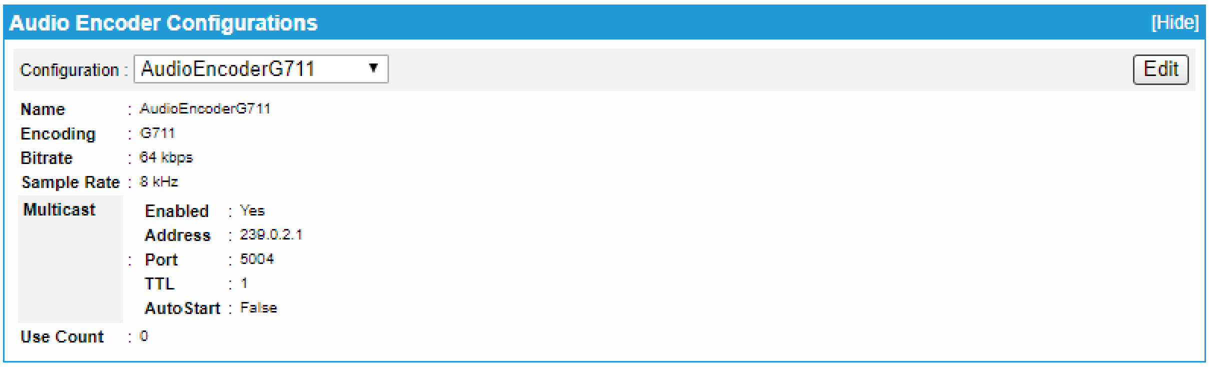
- Select an `AudioSourceConfiguration` object from the Configuration drop-down list in the upper area.

The lower area of the panel shows the properties of the selected object:

Audio source field description

Name	Description
Name	Name of the AudioSourceConfiguration object
Audio Source	
Name	Name of the audio source
Channels	Number of audio channels of the audio source
Use Count	Number of ONVIF Media Profiles using that AudioSourceConfiguration object

Audio Encoder Configurations panel



Audio Encoder Configurations [Hide]

Configuration : AudioEncoderG711 ▼ Edit

Name : AudioEncoderG711
Encoding : G711
Bitrate : 64 kbps
Sample Rate : 8 kHz
Multicast :
Enabled : Yes
Address : 239.0.2.1
Port : 5004
TTL : 1
AutoStart : False
Use Count : 0

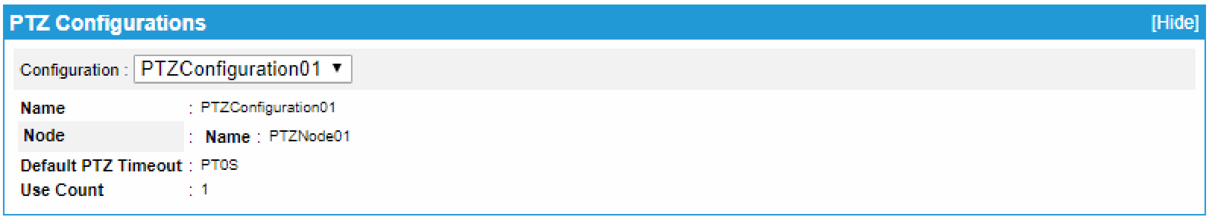
Audio Encoder Configurations panel

- Select an audio encoding method from the Configuration drop-down list in the upper area.
- Click Edit in the upper area of the panel to open the Audio Encoder Configuration Edition page.

Audio encoder field description

Name	Description
Name	Name of the `AudioEncoderConfiguration` object
Encoding	Used audio codec
Bitrate	Bit rate of the encoded audio stream
Sample Rate	Sampling rate of the encoded audio stream
Multicast	
Enabled	Indicates if the RTP multicast streaming of the encoded video is properly configured with a non-zero IP address and port number.
Address	IP address of the multicast group. *In IPv4, addresses 224.0.0.0 through 239.255.255.255 are designated as multicast addresses.*
Port	Port number of the multicast group
TTL	Time-To-Live of the multicast IP datagrams. *Usually 1 since the datagrams stops after the first router.*
AutoStart	Indicates the persistence of multicast streaming. *When true, the multicast streaming starts automatically.*
Use Count	Number of ONVIF Media Profiles using this object

PTZ Configurations panel



PTZ Configurations [Hide]

Configuration : PTZConfiguration01 ▼

Name : PTZConfiguration01

Node : Name : PTZNode01

Default PTZ Timeout : PT0S

Use Count : 1

PTZ Configurations panel

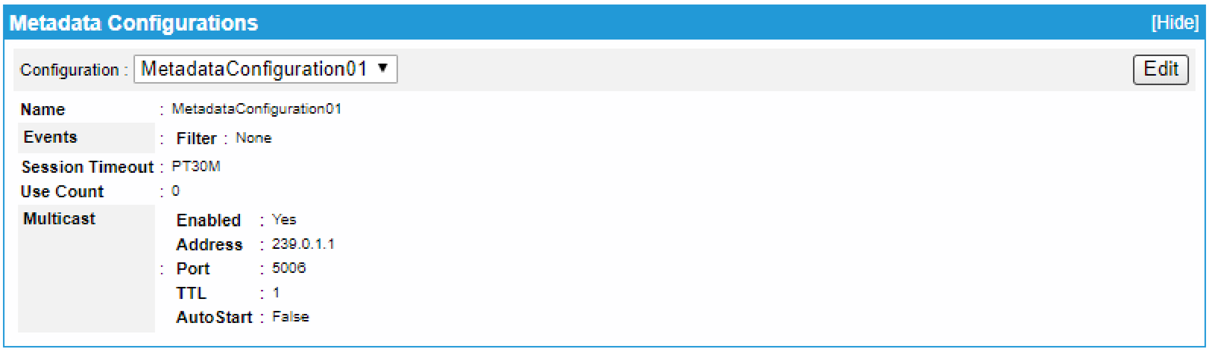
- Select a `PTZConfiguration` object from the Configuration drop-down list in the upper area.

The lower area of the panel shows the properties of the selected object:

PTZ field description

Name	Description
Name	Name of the PTZConfiguration object
Node - Name	Name of the PTZ node, e.g. PTZNode01
Default PTZ Timeout	Default timeout value for the continuous movements. *The duration is expressed using the W3C lexical representation: PnYn MnDTnH nMnS* .
Use Count	Number of ONVIF Media Profiles using that PTZConfiguration object.

Metadata Configurations panel


Metadata Configurations [Hide]

Configuration : MetadataConfiguration01 ▼ Edit

Name : MetadataConfiguration01

Events : Filter : None

Session Timeout : PT30M

Use Count : 0

Multicast : Enabled : Yes
Address : 239.0.1.1
Port : 5006
TTL : 1
AutoStart : False

Metadata Configurations panel

- Select a `MetadataConfiguration` object from the Configuration drop-down list in the upper area.

Metadata field description

Name	Description
Name	Name of the MetadataConfiguration object
Events - Filter	List of filtered event items. *When empty: means that no events are filtered.*
Session Timeout	RTSP session timeout. *The duration is expressed using the W3C lexical representation: PnYn MnDTnH nMnS*
Use Count	Number of ONVIF Media Profiles using that object
Multicast	
Enabled	Indicates if the RTP multicast streaming of the metadata is properly configured with a non-zero IP address and port number.
Address	IP address of the multicast group. *In IPv4, addresses 224.0.0.0 through 239.255.255.255 are designated as multicast addresses.*
Port	Port number of the multicast group.
TTL	Time-To-Live of the multicast IP datagrams. *Usually 1 since the datagrams stops after the first router.*
AutoStart	Indicates the persistence of multicast streaming. *When true, the multicast streaming starts automatically.*

4.1. Edit Video Encoder Configuration Page

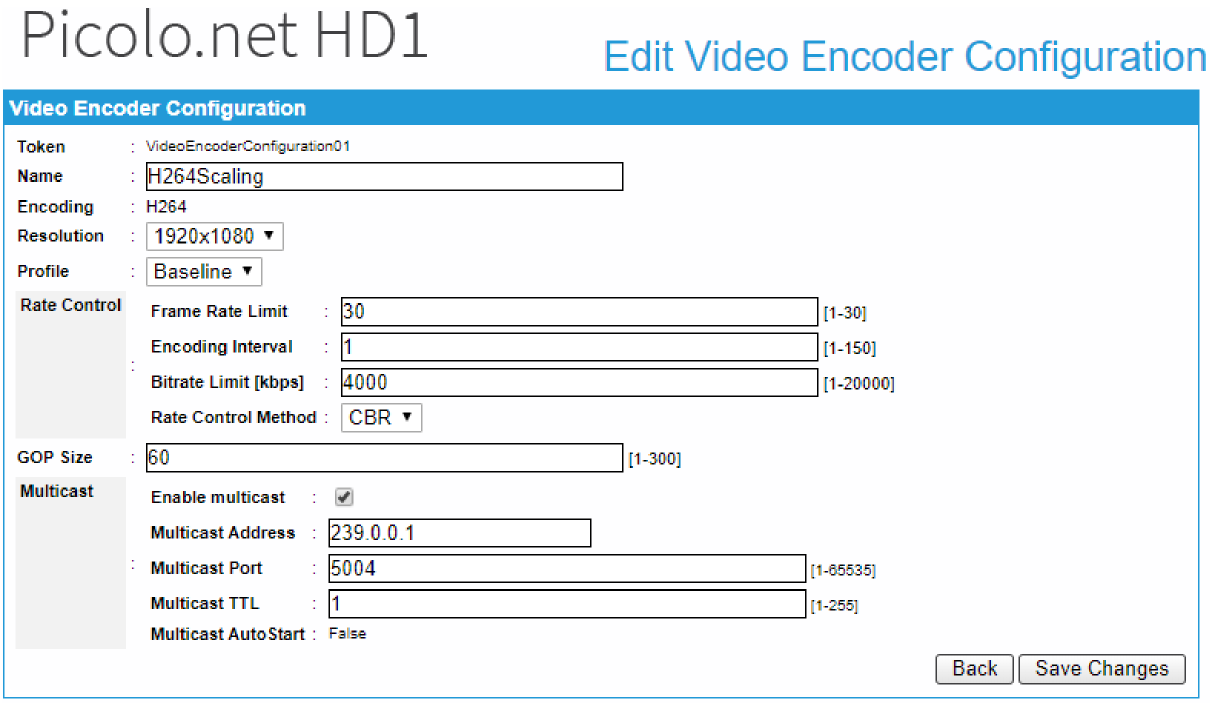
The Edit Video Encoder Configuration page allows the edition of the properties of the `VideoEncoderConfiguration` object.

It shows a single panel: the Video Encoder Configuration panel.

The layout of the Video Encoder Configurations panel is specific to the video encoding method.

Video Encoder Configuration Panels

H.264/H.265 video encoder configuration panel



Picolo.net HD1 — Edit Video Encoder Configuration

Video Encoder Configuration

Token : VideoEncoderConfiguration01

Name : H264Scaling

Encoding : H264

Resolution : 1920x1080 ▼

Profile : Baseline ▼

Rate Control :

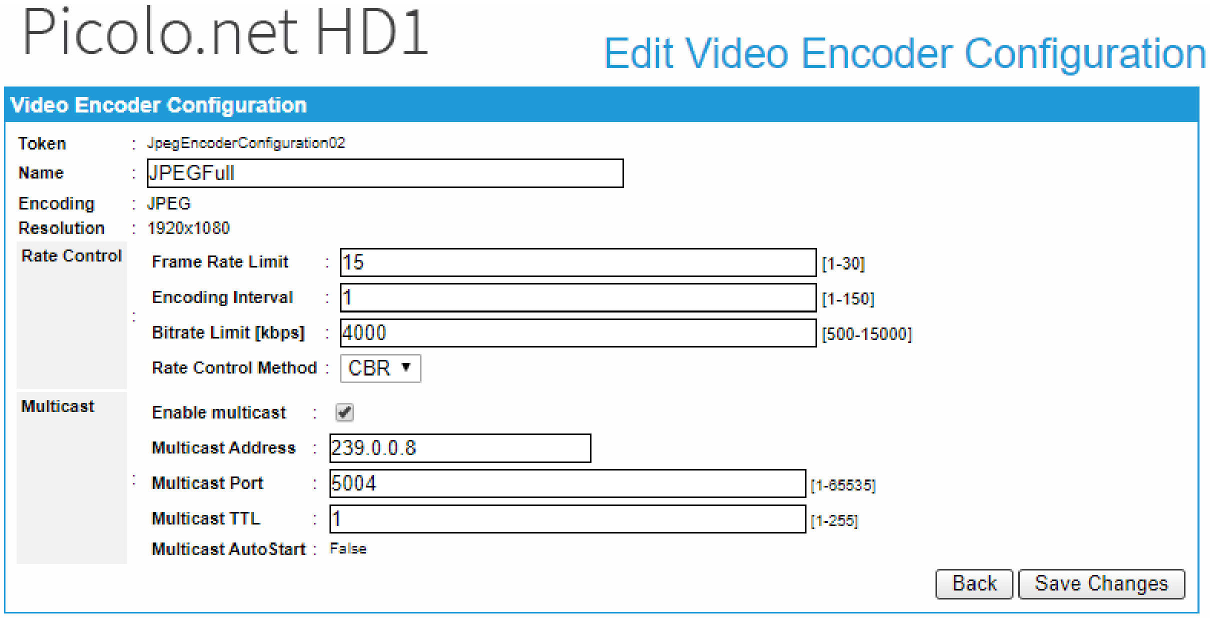
- Frame Rate Limit : 30 [1-30]
- Encoding Interval : 1 [1-150]
- Bitrate Limit [kbps] : 4000 [1-20000]
- Rate Control Method : CBR ▼

GOP Size : 60 [1-300]

Multicast :

- Enable multicast : ☑
- Multicast Address : 239.0.0.1
- Multicast Port : 5004 [1-65535]
- Multicast TTL : 1 [1-255]
- Multicast AutoStart : False

JPEG video encoder configuration panel



Picolo.net HD1

Edit Video Encoder Configuration

Video Encoder Configuration

Token	:	JpegEncoderConfiguration02		
Name	:	JPEGFull		
Encoding	:	JPEG		
Resolution	:	1920x1080		
Rate Control	:	Frame Rate Limit	15	[1-30]
		Encoding Interval	1	[1-150]
		Bitrate Limit [kbps]	4000	[500-15000]
		Rate Control Method	CBR ▾	
Multicast	:	Enable multicast	☑	
		Multicast Address	239.0.0.8	
		Multicast Port	5004	[1-65535]
		Multicast TTL	1	[1-255]
		Multicast AutoStart	False	

Video encoder configuration panel - common fields

Name	Description
Token	The token name of the `VideoEncoderConfiguration` object, e.g. `VideoEncoderConfiguration01`. *This field cannot be edited.*
Name	A friendly name given to the configuration. *Default value = token name*
Encoding	Video encoding method: H.264 or JPEG
Resolution	The resolution of the encoded image, e.g. 1920x1080
Rate Control - Frame Rate Limit	The maximum output frame rate of the encoded stream, in fps. *If an EncodingInterval is provided, the resulting encoded frame rate will be reduced by the given factor.*
Rate Control - Encoding Interval	The interval at which images are encoded and transmitted. *A value of 1 means that every frame is encoded, a value of 2 means that every 2nd frame is encoded,...*
Rate Control - Bitrate Limit	The maximum output bit rate in kbps. *This field cannot be edited in case of JPEG encoding.*
Multicast - Enable multicast	Check the box to configure RTP multicast streaming.
Multicast - Multicast Address	The IP address of the multicast group. *In IPv4, addresses 224.0.0.0 through 239.255.255.255 are designated as multicast addresses.*

Name	Description
Multicast - Multicast Port	The port number of the multicast group.
Multicast - Multicast TTL	The Time-To-Live of the multicast IP datagrams. *Usually 1 since the datagrams stops after the first router.*
Multicast - Multicast AutoStart	Indicates the persistence of multicast streaming. *When true, the multicast streaming starts automatically. This field cannot be edited.This is enabled/disabled by clicking on the Start/Stop Multicast button (in the Live Media Panel).*

Video encoder configuration panel - H.264 / H.265 specific fields

Name	Description
GOP Size	Length of the Group of Pictures (or Video frames). *Determines typically the interval in which the I-Frames will be coded. An entry of 1 indicates I-Frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-Frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames.*
Profile	The H.264 encoder profiles: baseline, main, or high.
Rate Control - Rate Control Method	The rate control method of the H.264 encoder. Possible values: ● VBR: Variable Bit Rate ● CBR: Constant Bit Rate
Low Latency	Check the box to configure the low-latency encoding method.

4.2. Edit Audio Encoder Configuration Page

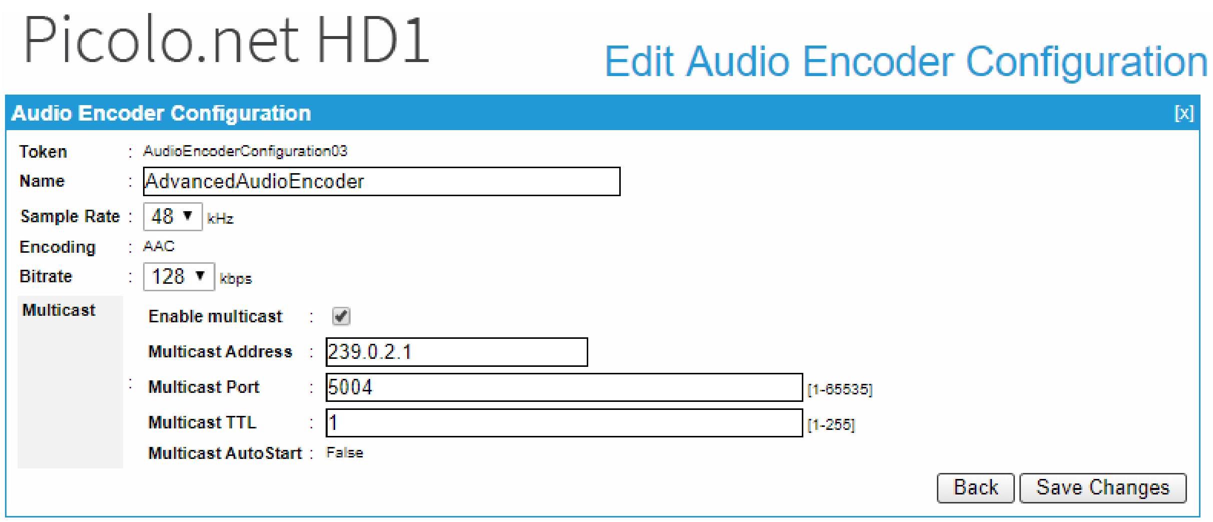
The Edit AudioEncoder Configuration page allows the edition of the properties of the `AudioEncoderConfiguration` object.

It shows a single panel: the Audio Encoder Configuration panel.

The layout of the Audio Encoder Configurations panel is specific to the audio encoding method.

Audio Encoder Configuration Panel

AAC Audio Encoder Configuration Panel



Picolo.net HD1 Edit Audio Encoder Configuration

Audio Encoder Configuration [x]

Token : AudioEncoderConfiguration03

Name : AdvancedAudioEncoder

Sample Rate : 48 ▼ kHz

Encoding : AAC

Bitrate : 128 ▼ kbps

Multicast :

Enable multicast : ☑

Multicast Address : 239.0.2.1

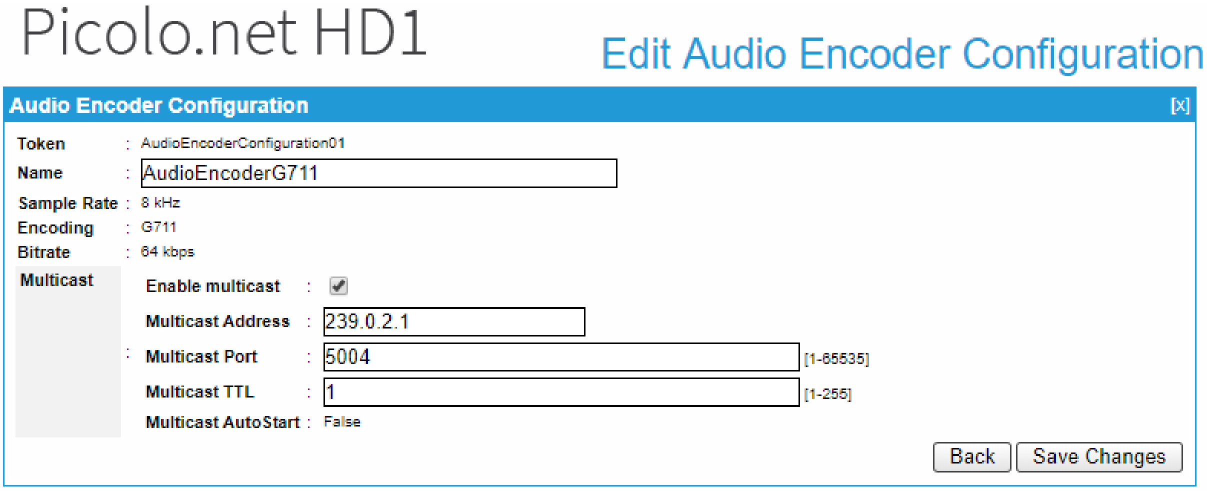
Multicast Port : 5004 [1-65535]

Multicast TTL : 1 [1-255]

Multicast AutoStart : False

Edit Audio Encoder Configuration panel - AAC case

G.711 audio encoder configuration panel



Picolo.net HD1

Edit Audio Encoder Configuration

Audio Encoder Configuration [x]

Token : AudioEncoderConfiguration01

Name : AudioEncoderG711

Sample Rate : 8 kHz

Encoding : G711

Bitrate : 64 kbps

Multicast :

Enable multicast : ☑

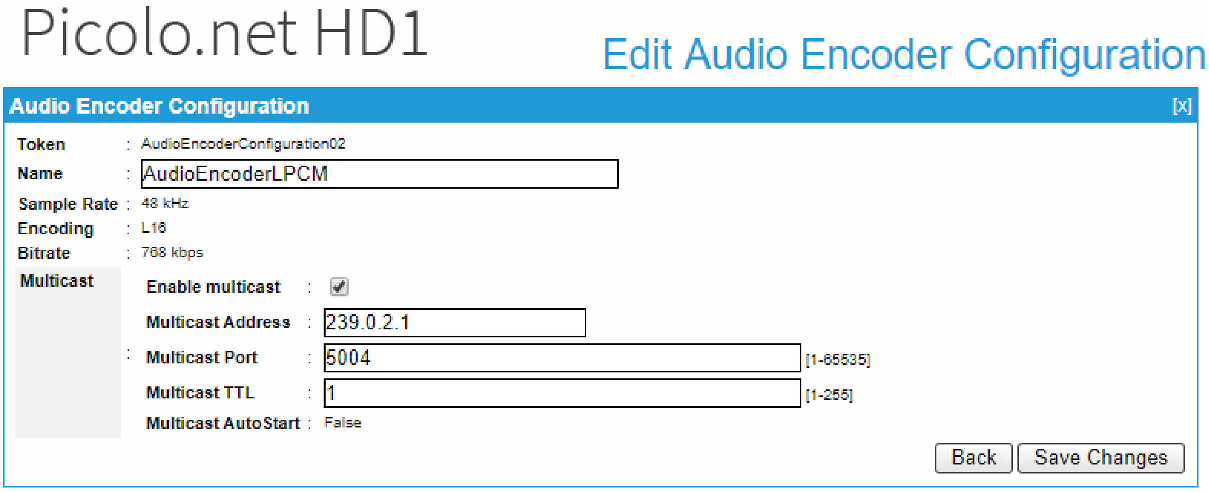
Multicast Address : 239.0.2.1

Multicast Port : 5004 [1-65535]

Multicast TTL : 1 [1-255]

Multicast AutoStart : False

LPCM audio encoder configuration edition panel



Picolo.net HD1

Edit Audio Encoder Configuration

Audio Encoder Configuration [x]

Token : AudioEncoderConfiguration02

Name : AudioEncoderLPCM

Sample Rate : 48 kHz

Encoding : L16

Bitrate : 768 kbps

Multicast :

Enable multicast : ☑

Multicast Address : 239.0.2.1

Multicast Port : 5004 [1-65535]

Multicast TTL : 1 [1-255]

Multicast AutoStart : False

Audio encoder field description

Name	**Description**
Token	Token name of the AudioEncoderConfiguration object, e.g. AudioEncoderConfiguration01. *This field cannot be edited.*
Name	User-Friendly name given to the configuration. *Default value = token name*
Encoding	Audio encoding method: ● *AAC: 1Advanced Audio Coding* ● *G711: G.711 μ-Law* ● *L16: 16-bit linear PCM*

Name	Description
Bitrate	Bitrate of the encoded audio stream expressed in kilobits per second. - *128 kbps for the AAC encoding method* - *64 kbps for the G.711 encoding method* - *768 kbps for the L16 encoding method*
Sample Rate	Sampling rate of the encoded audio stream expressed in kHz. - Select 48 kHz for the AAC encoding method - Select 8 kHz for the G.711 encoding method (Default setting) - Select 48 kHz for the L16 encoding method
Multicast - Enable multicast	Check the button to configure RTP multicast streaming.
Multicast - Multicast Address	IP address of the multicast group. *In IPv4, addresses 224.0.0.0 through 239.255.255.255 are designated as multicast addresses.*
Multicast - Multicast Port	Port number of the multicast group.
Multicast - Multicast TTL	Time-To-Live of the multicast IP datagrams. *Usually 1 since the datagrams stops after the first router.*
Multicast - Multicast AutoStart	Indicates the persistence of multicast streaming. *When true, the multicast streaming starts automatically. This field cannot be edited.This is enabled/disabled by clicking on the Start/Stop Multicast button (in the Live Media Panel).*

4.3. Edit Metadata Configuration Page

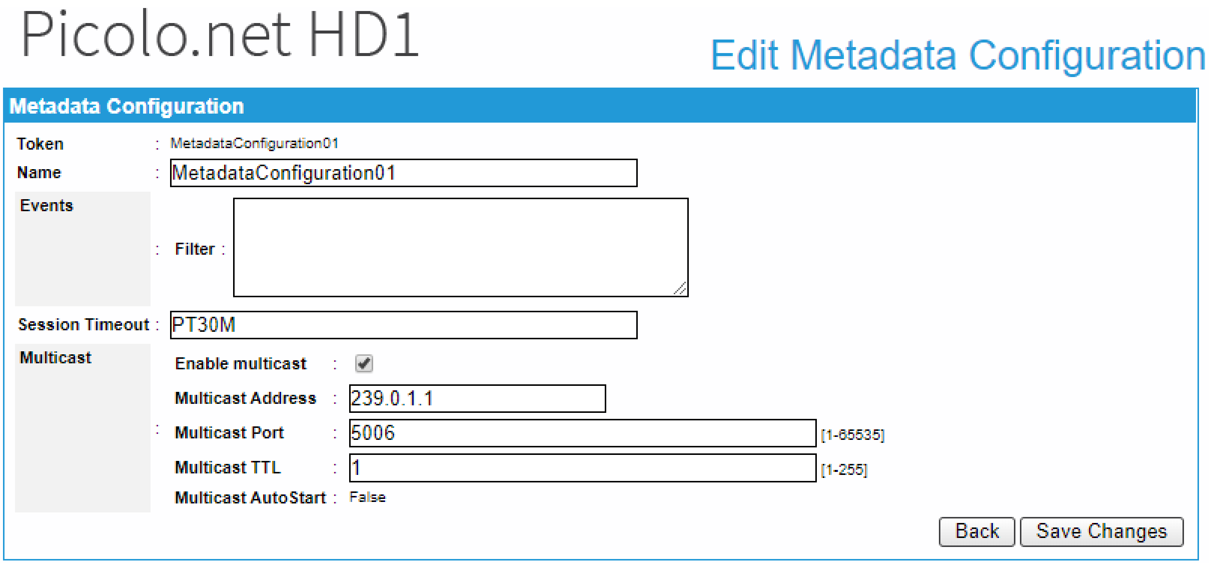
NOTE

The features associated to this configuration page are not currently available.

The Edit Metadata Configuration page allows the edition of the properties of the `MetadataConfiguration` object.

It shows a single panel: the Metadata Configuration panel.

Metadata Configuration Edition panel



Token : MetadataConfiguration01

Name : MetadataConfiguration01

Events : Filter :

Session Timeout : PT30M

Multicast :

Enable multicast : ☑

Multicast Address : 239.0.1.1

Multicast Port : 5006 [1-65535]

Multicast TTL : 1 [1-255]

Multicast AutoStart : False

Edit Metadata Configuration panel

5. Digital Inputs & Relay Outputs Page

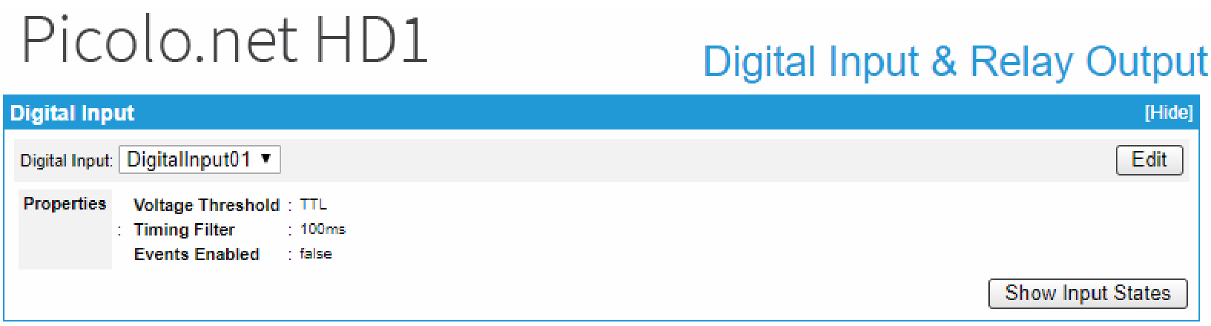
NOTE
The features associated to this configuration page are not currently available.

The Digital Inputs & Relay Outputs page allows the user to view or edit the configuration of `DigitalInput` and `RelayOutput` objects.

A `DigitalInput` object represents an Alarm Input port; a `RelayOutput` object represents one Relay Output port.

The panels composing this pane can be hidden or shown individually by clicking on the [Hide] or [Show] text.

Digital Input Panel



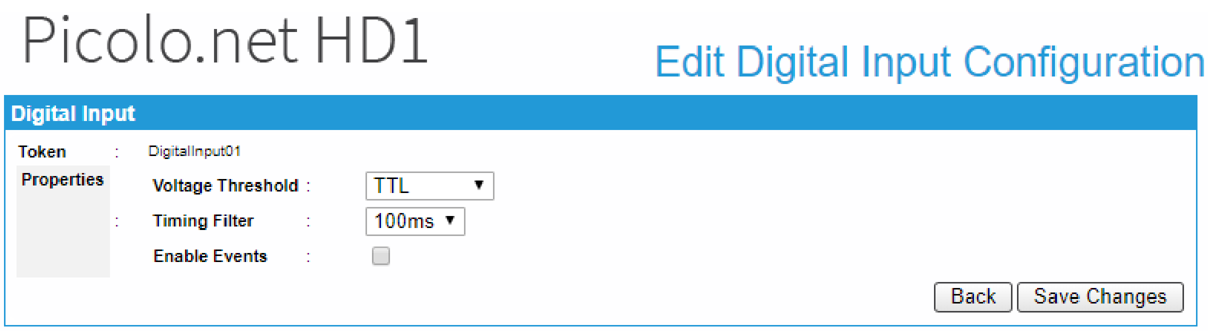
Digital Input: DigitalInput01 ▼

Properties:

Voltage Threshold	:	TTL
Timing Filter	:	100ms
Events Enabled	:	false

Digital Input panel

EDIT Digital Input Configuration Page



Token : DigitalInput01

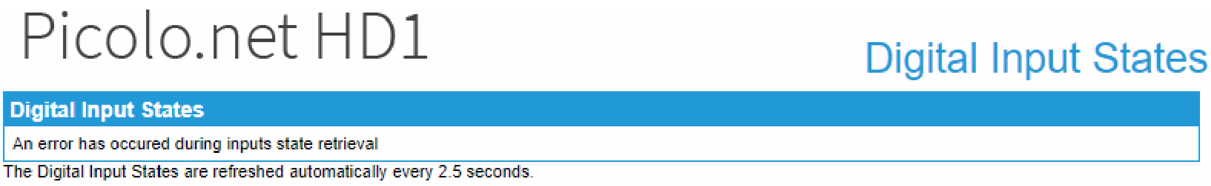
Properties:

Voltage Threshold	:	TTL ▼
Timing Filter	:	100ms ▼
Enable Events	:	□

Edit Digital Input panel

Digital Inputs States page

The Digital Inputs States page displays a single panel allowing the user to view the state of all `DigitalInput` objects.



Picolo.net HD1

Digital Input States

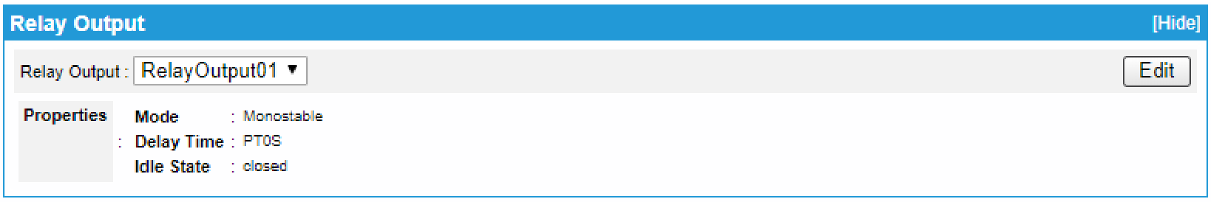
Digital Input States

An error has occured during inputs state retrieval

The Digital Input States are refreshed automatically every 2.5 seconds.

Digital Input States panel

Relay Output Panel



Relay Output [Hide]

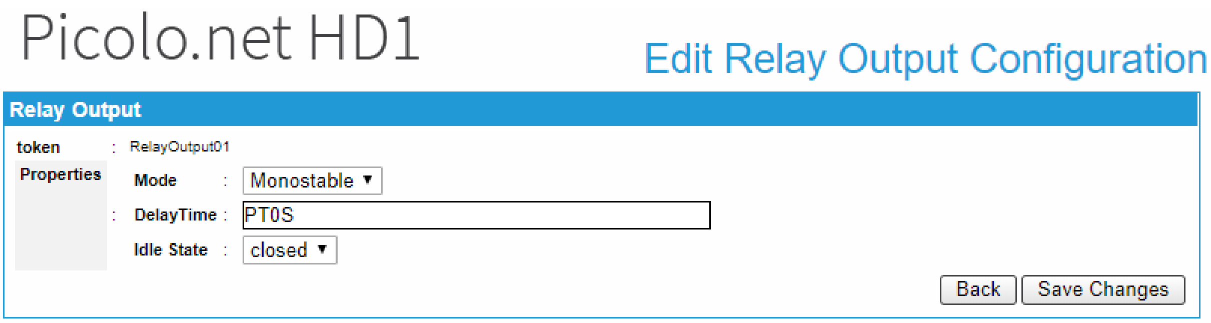
Relay Output : RelayOutput01 ▼ Edit

Properties :

Mode	:	Monostable
Delay Time	:	PT0S
Idle State	:	closed

Relay Output panel

Edit Relay Output Configuration Page



Picolo.net HD1

Edit Relay Output Configuration

Relay Output

token : RelayOutput01

Properties :

Mode	:	Monostable ▼
DelayTime	:	PT0S
Idle State	:	closed ▼

Back Save Changes

Edit Relay Output panel

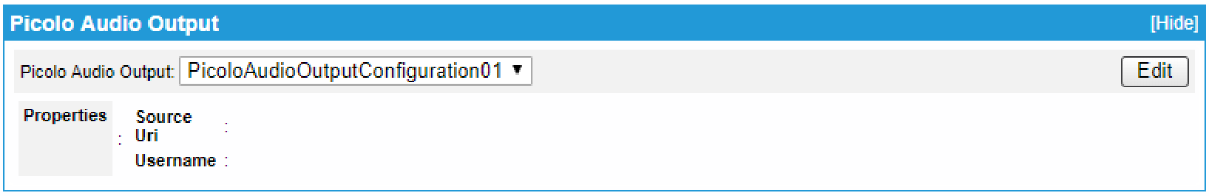
6. Audio Outputs Page

NOTE
The features associated to this configuration page are not currently available.

The Audio Outputs page allows the user to view or edit the configuration of `Picolo AudioOutput` objects.

A `Picolo AudioOutput` object represents one Audio Output port.

Picolo Audio Output Panel



Picolo Audio Output

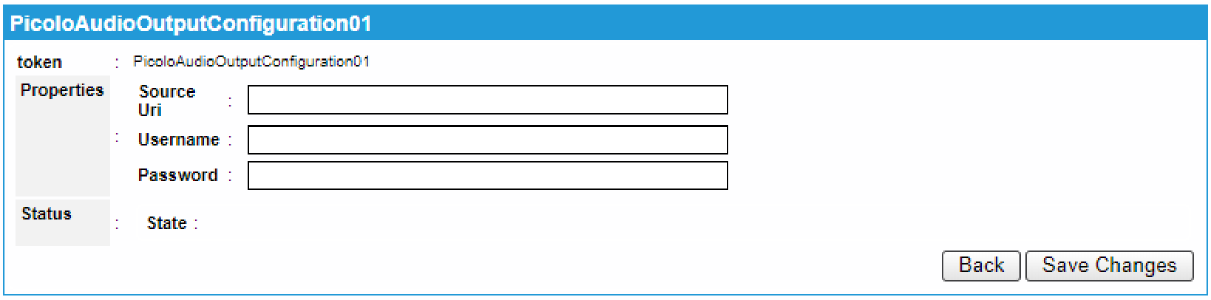
Picolo Audio Output: PicoloAudioOutputConfiguration01

Properties : Source Uri :

Username :

Picolo Audio Output Panel

Edit Picolo Audio Output Configuration page



PicoloAudioOutputConfiguration01

token : PicoloAudioOutputConfiguration01

Properties : Source Uri :

Username :

Password :

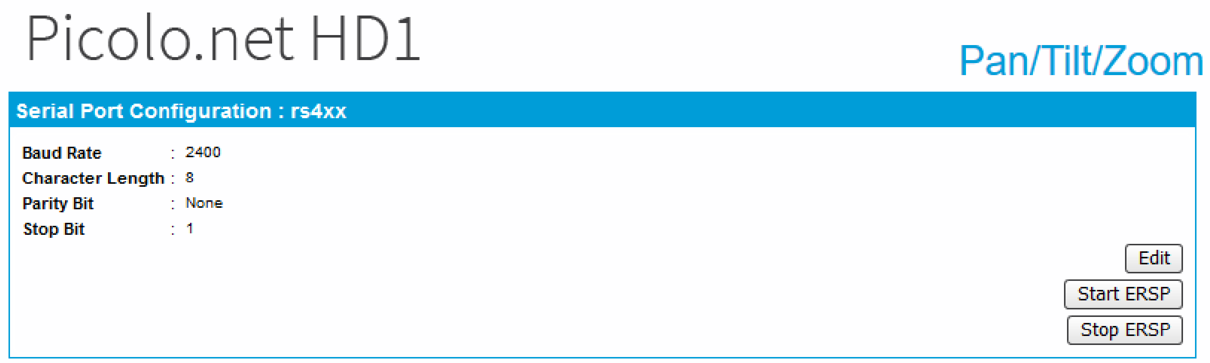
Status : State :

7. PTZ Page

The PTZ page allows the user to view or edit the configuration of the serial ports and the `PTZNode` objects.

See also the section "Using the Euresys Remote Serial Protocol (ERSP)" on page 99.

Serial Port Configuration panel



Serial Port Configuration : rs4xx

Baud Rate	:	2400
Character Length	:	8
Parity Bit	:	None
Stop Bit	:	1

Serial Port Configuration panel

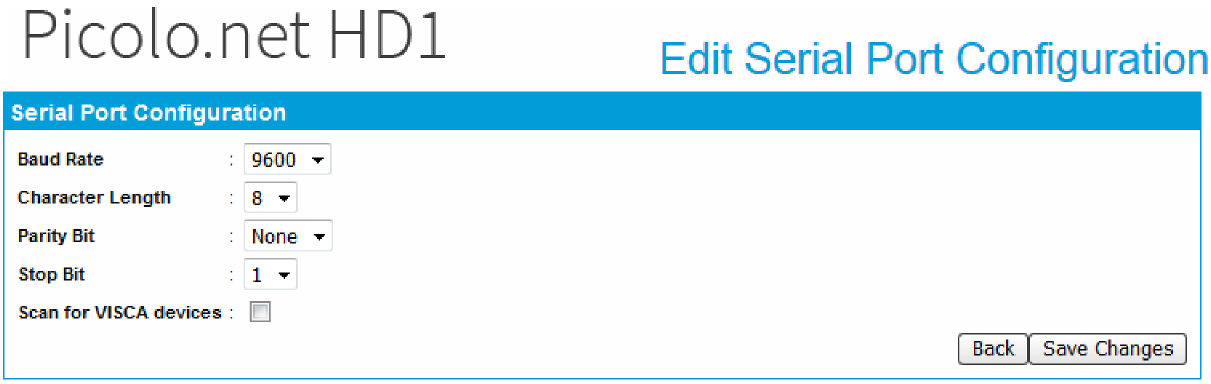
The Serial Port Configuration panel allows the user to view the properties of RS-232 and RS-4xx (Pelco) serial ports.

Serial Port Configuration panel fields description

Name	Description
Baud Rate	The baud rate of the serial port
Character Length	Number of bits per character
Parity Bit	Presence and polarity of the parity bit
Stop Bit	Number of stop bits

Clicking on the Edit button opens the Edit Serial Port Configuration page.

Edit Serial Port Configuration Page



Picolo.net HD1

Edit Serial Port Configuration

Serial Port Configuration

Baud Rate : 9600

Character Length : 8

Parity Bit : None

Stop Bit : 1

Scan for VISCA devices : □

Edit Serial Port Configuration panel

The Edit Serial Port Configuration page displays a single panel, named Serial Port Configuration, that allows you to edit the properties of the serial port.

Click on the Save Changes button in the lower right area to save your settings.

Focus control for VISCA cameras

For cameras supporting the Sony VISCA protocol, you can control the camera focus automatically or manually over the RS-232 connector using the ONVIF Imaging service 2.0.

About this feature:

- To enable VISCA support, check the Scan for VISCA devices feature in the PTZ/serial configuration web page.
- When this feature is activated, the Picolo.net HD1 sends VISCA camera discovery messages at boot time.
- If the Picolo.net HD1 cannot find any camera during the scan at boot, it can run another scan every time you call the Imaging Service.
- When this feature is not activated, the Picolo.net HD1 does not send any message over the RS-232 port unless you explicitly request it.

Focus control for Pelco cameras

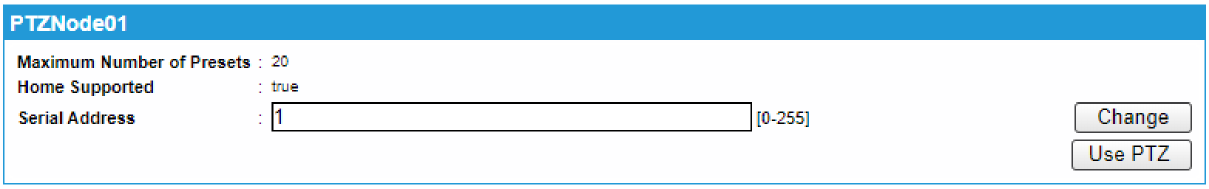
For cameras supporting the Pelco-D protocol, you can control the camera focus automatically or manually over the RS-485 connector using the ONVIF Imaging service 2.0.

This feature is automatically enabled unless you enable the VISCA camera support.

Because the Pelco-D protocol does not have a way to report the current position of the focus:

- You can only use the **continuous** and the **relative** ONVIF modes.
- The system emulates the **relative** mode by inserting a delay between the “move focus” and the “stop focus move” commands.

PTZNode panel



PTZNode01

Maximum Number of Presets : 20
Home Supported : true
Serial Address : 1 [0-255]

PTZNode panel

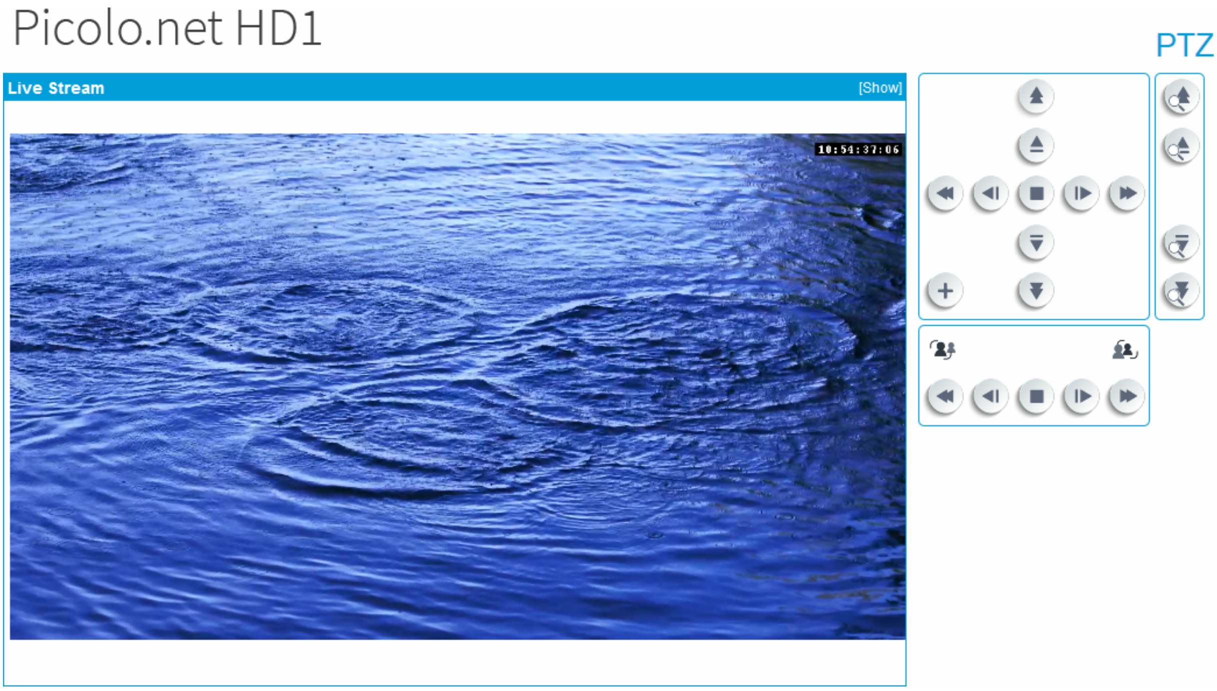
The PTZNode panel allows the user to view the properties of the corresponding `PTZNode`:

PTZNode panel fields description

Name	Description
Maximum Number of Presets	Indicates the maximum number of presets supported by the PTZ protocol. *20 for Pelco-D protocol.*
Home Supported	Indicates if the home command is supported by the PTZ protocol. *True for Pelco-D protocol.*
Serial Address	The address given to the PTZ node

Clicking on the Change button assigns the serial address to the PTZ node.

Clicking on the Use PTZ button adds PTZ controls for this PTZ node.



Live Stream panel with PTZ controls

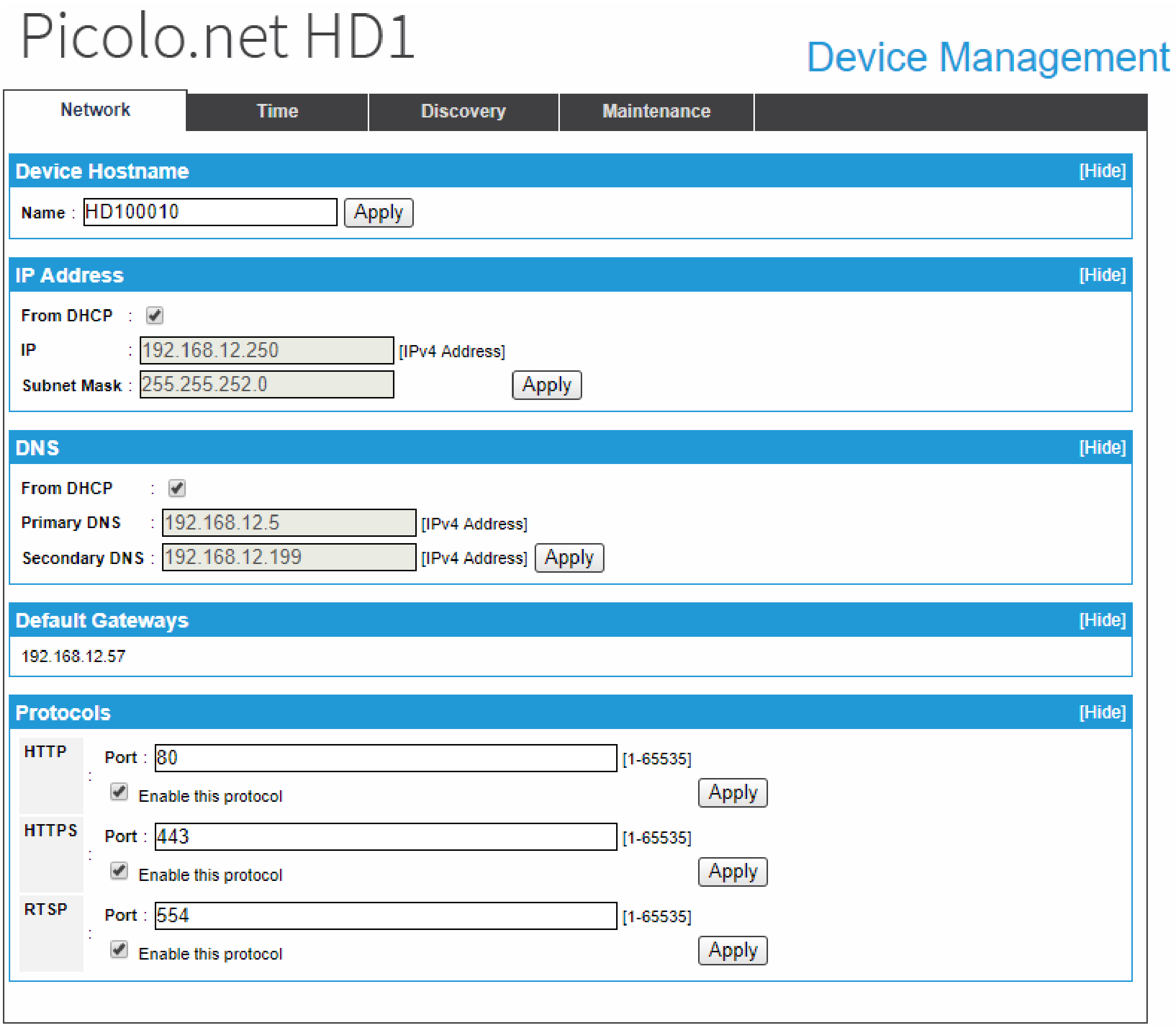
The Add button in the PTZ controls allows the user to recording the current PTZ position as a preset in the camera.

A third-party software is still required to update or delete such presets.

8. Device Management Page

8.1. Network Tab

The Network tab of the Device Management page allows the user to view or edit all the network related settings.



Device Hostname [Hide]

Name : HD100010 Apply

IP Address [Hide]

From DHCP : ☑

IP : 192.168.12.250 [IPv4 Address]

Subnet Mask : 255.255.252.0 Apply

DNS [Hide]

From DHCP : ☑

Primary DNS : 192.168.12.5 [IPv4 Address]

Secondary DNS : 192.168.12.199 [IPv4 Address] Apply

Default Gateways [Hide]

192.168.12.57

Protocols [Hide]

HTTP : Port : 80 [1-65535]
☑ Enable this protocol Apply

HTTPS : Port : 443 [1-65535]
☑ Enable this protocol Apply

RTSP : Port : 554 [1-65535]
☑ Enable this protocol Apply

Network tab

Device Host Name Panel

- The Device Hostname panel allows the user to view and/or edit the device host name.
- Clicking on the Apply button registers the change. It will be effective after a device reboot.

IP Address Panel

- The IP Address panel allows the user to view and/or edit the device IP address and the sub-net mask.
- When the From DHCP check box is checked, the IP address is obtained automatically using DHCP. The IP and Subnet Mask fields reflect the values assigned automatically by the DHCP server. These values cannot be modified.
- When the From DHCP check box is unchecked, the user is allowed to change the IP and Subnet Mask fields.
- Clicking on the Apply button registers the change. It will be effective after a device reboot.

DNS Panel

- The DNS panel allows the user to view and/or edit the IP address of the primary and secondary DNS servers.
- When the From DHCP check box is checked, the IP addresses of the primary and secondary DNS servers are obtained automatically using DHCP. The Primary DNS and Secondary DNS fields reflect the values assigned automatically by the DHCP server. These values cannot be modified.
- When the From DHCP check box is unchecked, the user is allowed to change the Primary DNS and Secondary DNS fields.
- Clicking on the Apply button registers the change. It will be effective after a device reboot.

Default Gateways Panel

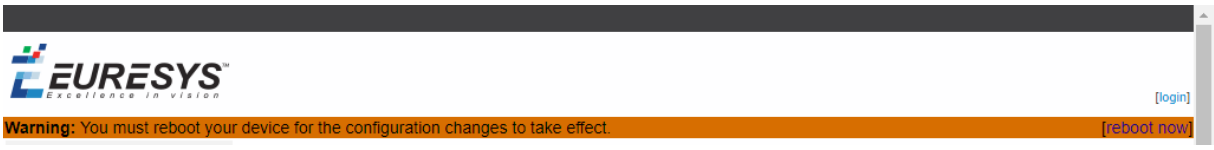
- The Default Gateways panel allows the user to view the IP address of the default gateways. When the IP address of the device is statically assigned, default gateways can be added, edited, or deleted.

Protocols Panel

- The Protocols panel allows the user to Individually enable/disable the HTTP, HTTPS, and RTSP protocols and assign a port number to each.

IP Change Panel

- After a change of the IP Adress settings, the IP Change panel is displayed. It indicates that the IP address change will be effective only after rebooting the device.
- Clicking on the OK button returns to the last page. The Must reboot banner appears on top of it:

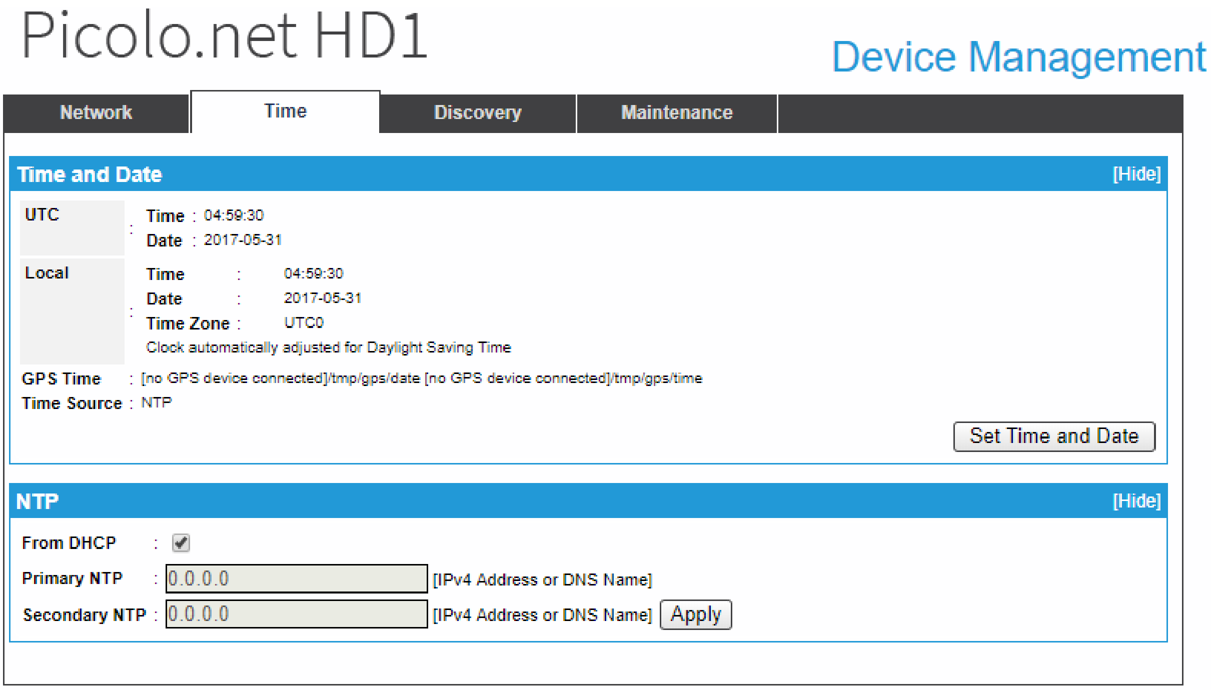


Warning: You must reboot your device for the configuration changes to take effect.

Must reboot banner

8.2. Time Tab

The Time tab of the Device Management page allows the user to view or edit all the time and date related settings.



Picolo.net HD1 Device Management

Time and Date [Hide]

UTC : Time : 04:59:30
Date : 2017-05-31

Local : Time : 04:59:30
Date : 2017-05-31
Time Zone : UTC0
Clock automatically adjusted for Daylight Saving Time

GPS Time : [no GPS device connected]/tmp/gps/date [no GPS device connected]/tmp/gps/time
Time Source : NTP

NTP [Hide]

From DHCP : ☑
Primary NTP : 0.0.0.0 [IPv4 Address or DNS Name]
Secondary NTP : 0.0.0.0 [IPv4 Address or DNS Name]

Time tab

Time and Date Panel

Name	Description
UTC - Time	The UTC (Coordinate Universal Time) time value.
UTC - Date	The UTC (Coordinate Universal Time) date value.
Local - Time	The local time value.
Local - Date	The local date value. *Expressed in YYYY-MM-DD format.*
Local - Time Zone	The local time zone rule. *Expressed in POSIX.1 TZ string format.*
GPS Time	The UTC time provided by the GPS receiver device.
Time Source	The source used for time and date synchronization.

- The Time and Date panel allows the user to view the time and date settings.

- The Time fields use the HH:MM:SS format. The Date fields use the YYYY-MM-DD format. The Time Zone field use the POSIX.1 TZ string format.
- A Clock automatically adjusted for Daylight Saving Time message indicates that the DST rule of the POSIX.1 TZ string is effectively considered by the Operating Systems.
- A Clock not automatically adjusted for Daylight Saving Time. message indicates that the DST rule of the POSIX.1 TZ string is ignored by the Operating Systems.
- Clicking on the Set Time and Date button opens the Edit Date and Time page.
- When the automatic GPS with NTP method is used, Time Source reports either `GPS` or `NTP` depending on whether a GPS device is actively providing time information.

Date and Time Panel

- The Date and Time panel of the Edit Date and Time page allows the user to modify all the time and date settings.
- The Time Source drop-down box allows to select the source of the time synchronization.
  - NTP selects the automatic synchronization method using NTP protocol.
  - NTP + GPS enables USB GPS device to act as an NTP time source.
  - Manual selects the manual synchronization method.

When manual synchronization method is selected, all the six fields of UTC Time area must be properly filled with the actual values of the UTC time.

- The drop-down box in the Time Zone area provides a list of time zone sorted by increasing UTC offset values. Selecting an item automatically fills the edit box with the corresponding POSIX.1 TZ string.

The validity of the TZ rules is not guaranteed. Indeed, TZ rules are subject to modification by civil authorities.

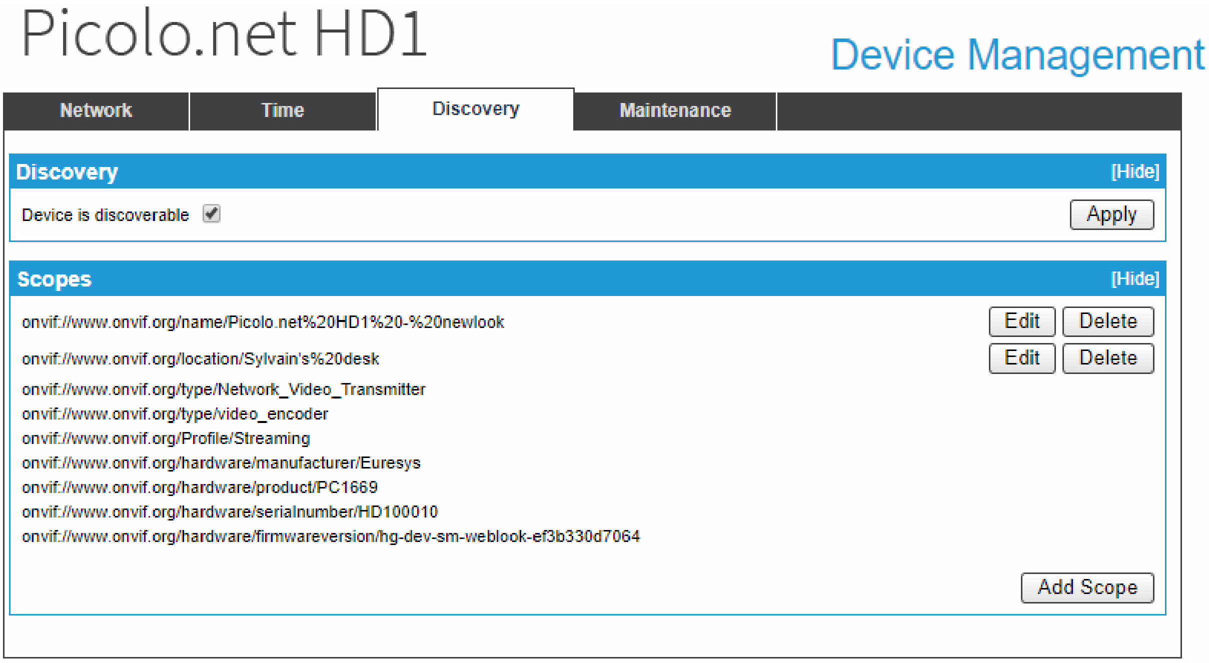
- The edit-box in the Time Zone area specifies the time zone rule expressed in POSIX.1 TZ string format. An empty field means that the local time is equal to the UTC time.
- The *Automatically adjust clock for Daylight Saving Time* check box controls the application of the DST (Daylight Savings Time) rule embedded in the time zone string. When checked, the device updates automatically the local time according to the DST rule. When unchecked, the device ignores the DST rule.
- Clicking on the Apply button immediately applies the settings.

NTP Panel

- The NTP panel allows the user to view and/or edit the IP address of the primary and secondary NTP servers.
- When the From DHCP check box is checked, the IP addresses of the primary and secondary NTPservers are obtained automatically using DHCP. The Primary NTP and Secondary NTP fields reflect the values assigned automatically by the DHCP server. These values cannot be modified.
- When the From DHCP check box is unchecked, the user is allowed to change the Primary NTP and Secondary NTP fields.
- Clicking on the Apply button registers the change.

8.3. Discovery Tab

The Discovery tab of the Device Management page allows the user to view or edit all the device discovery settings.



Device is discoverable ☑

onvif://www.onvif.org/name/Picolo.net%20HD1%20-%20newlook
onvif://www.onvif.org/location/Sylvain's%20desk
onvif://www.onvif.org/type/Network_Video_Transmitter
onvif://www.onvif.org/type/video_encoder
onvif://www.onvif.org/Profile/Streaming
onvif://www.onvif.org/hardware/manufacturer/Euresys
onvif://www.onvif.org/hardware/product/PC1669
onvif://www.onvif.org/hardware/serialnumber/HD100010
onvif://www.onvif.org/hardware/firmwareversion/hg-dev-sm-weblook-ef3b330d7064

Discovery tab

Discovery Panel

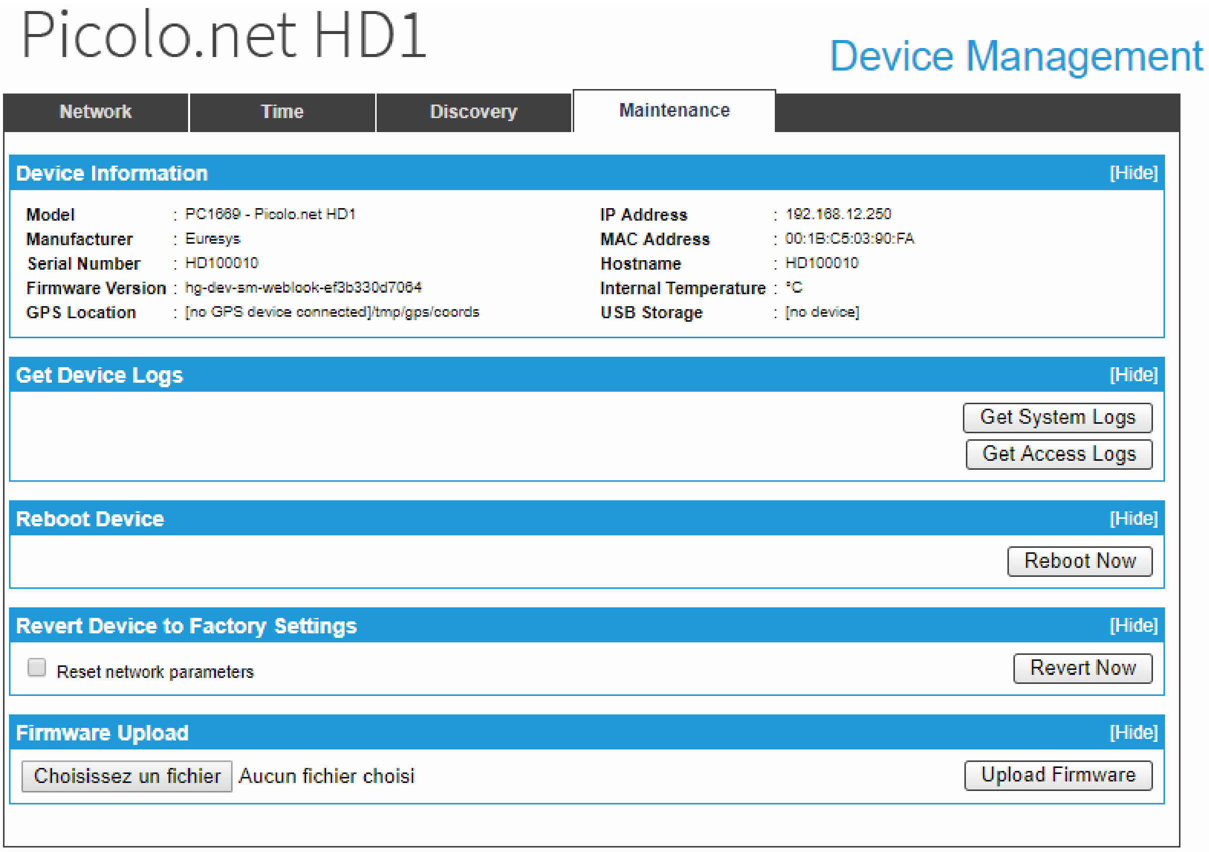
- The Device is discoverable check box controls the ability to discover the device on the network using the discovery functions of the ONVIF Device Web Service. When checked, the device is discoverable. When unchecked, the device don't reply to the discovery request messages.
- Clicking on the Apply button applies immediately the settings.

Scopes Panel

- The Scopes panel allows the user to view and create ONVIF device scopes.
- Clicking on the Add Scope button opens a dialog box allowing to create a new scope.
- For user editable scopes, the panel provides an Edit button and a Delete button. Clicking on an Edit button opens a dialog box allowing to modify the scope. Clicking on a Delete button opens a dialog box allowing to delete the scope.

8.4. Maintenance Tab

The Maintenance tab of the Device Management page allows the user to perform maintenance tasks.



Model : PC1669 - Picolo.net HD1
Manufacturer : Euresys
Serial Number : HD100010
Firmware Version : hg-dev-sm-weblook-ef3b330d7064
GPS Location : [no GPS device connected]/tmp/gps/coords
IP Address : 192.168.12.250
MAC Address : 00:1B:C5:03:90:FA
Hostname : HD100010
Internal Temperature : °C
USB Storage : [no device]

□ Reset network parameters

Maintenance tab

Device Information Panel

Name	Description
Model	Product code and product name of the device
Manufacturer	Manufacturer name of the device
Serial Number	Serial number of the device
Firmware Version	Major and minor firmware version numbers of the device
GPS Location	GPS coordinates of the actual device location
IP Address	IPv4 address of the device currently assigned to the device

Name	Description
MAC Address	MAC Address of the LAN port of the device
Hostname	Host name currently assigned to the device
Internal Temperature	Internal temperature of the device, expressed in °C
USB Storage	Indication of presence and capacity of an USB 2.0 compatible mass storage device (if any).

A [no GPS device connected] message indicates that there are no active GPS device attached to a USB port.

A no device message indicates that there are no storage device attached to a USB port.

Get Device Logs Panel

- The Get Device Logs panel allows the user to retrieve log files from the device.
- Clicking on the Get Systems Logs button initiates the download of the `system.logs.tar.gz file` containing the system logs data.
- Clicking on the Get Access Logs button initiates the download of the `access.logs.tar.gz file` containing the access logs data.

In the log files, time is expressed in UTC time.

NOTE
In case of video pipeline crash recovered by the watchdog, the memento dump is collected and added to the system logs. In this case, a warning message displayed at the top of the web interface will notify you.

Since the device can store a single memento dump, get the system logs to free up space for future logs.

Reboot Device Panel

- The Reboot Device panel allows the user to reboot the device.
- Clicking on the Reboot Now button opens a dialog box allowing to initiate or cancel the task.

Revert Device to Factory Settings Panel

- The Revert Device to Factory Settings panel allows the user to revert the device settings to their initial value at factory output.
- Clicking on the Revert Now button opens a dialog box allowing to initiate or cancel the task.
- The Reset network parameters check box controls the reverting of the network settings and the user database. When checked, the network related settings and the user database are also reverted. When unchecked, the network related settings and the user database are not reverted.

Firmware Upload Panel

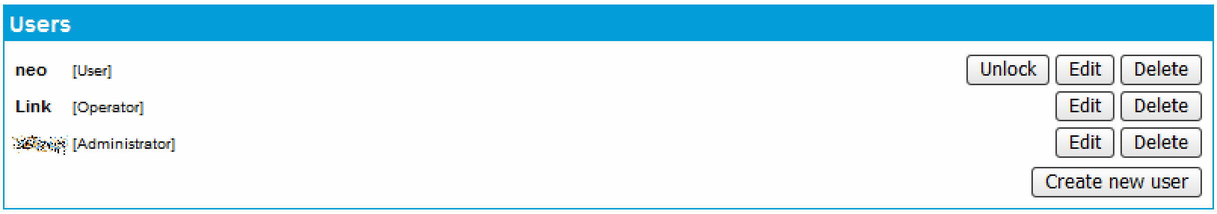
- The Firmware Upload panel allows the user to upload a firmware to the device.
- Clicking on the Browse button opens the file browser for example Windows Explorer allowing to select the firmware image file to upload.
- Clicking on the Upload Firmware opens a dialog box allowing to initiate or cancel the task.

9. Users Management Page

The Users Management page allows a user (with sufficient rights) to create and delete users and to view or edit user properties.

Users panel

- The Users panel of the Users Management page displays the list of users.
- Each list item contains the user name and the user level between square brackets, an Edit and a Delete button and, in specific cases, an Unlock button.
    - Click the Delete button to delete the user.
    - Click the Edit button or the Create New User button to open the User Edition page.
    - If a user is locked after repeated login failures and you are an administrator, click the Unlock button to re-activate the account before the timer expiration.



Users

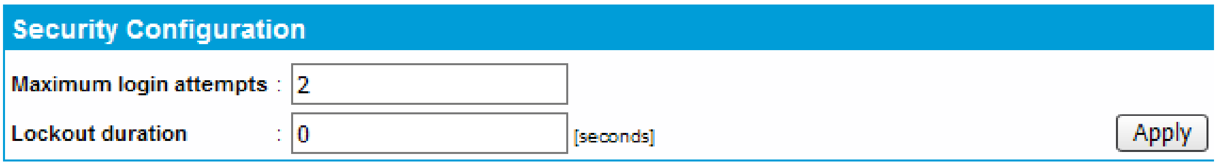
neo [User]

Link [Operator]

[redacted] [Administrator]

Security Configuration panel

- The Security Configuration panel is displayed only for administrators.
- It allows adjusting the parameters of access controls such as the auto-lockout on repeated login failures.
  - □ Maximum login attempts before auto-lockout triggers.
  - □ Lockout duration in seconds, from the first failed login until the user is allowed to attempt login again.

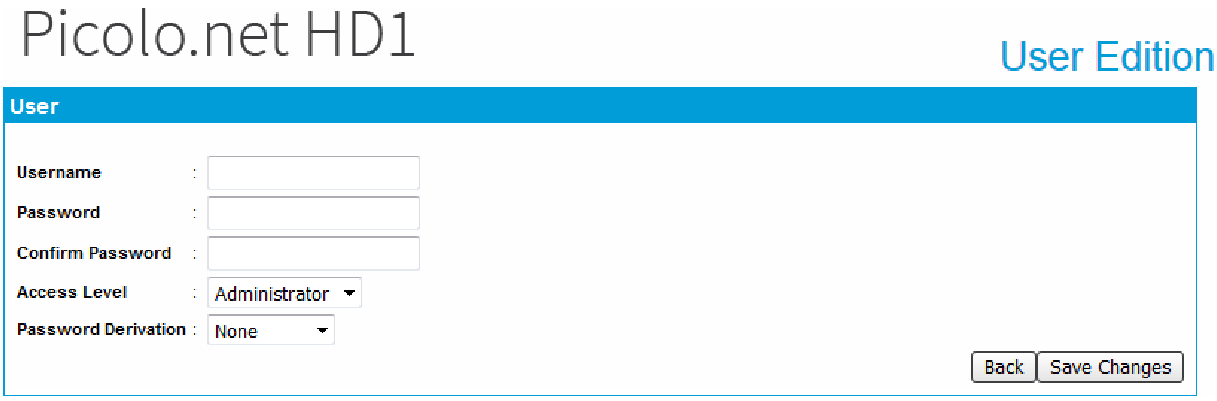


Security Configuration

Maximum login attempts : 2

Lockout duration : 0 [seconds]

User panel on the User Edition page



Picolo.net HD1

User Edition

User

Username :

Password :

Confirm Password :

Access Level : Administrator

Password Derivation : None

User panel

User panel fields description

Name	Description
Username	User name
Password	User password
Confirm password	User password again
Access Level	User access level. Possible values: Administrator, Operator, User
Password Derivation	Enable, disable and configure ONVIF password derivation.

Password Derivation

Password derivation allows the user of multiple devices to type the same string when authenticating on any device while the value stored on the device is actually different for each device.

Value	Description
None	No password derivation, the device password is directly typed by the user. Default setting.
Onvif 1.0	The password is computed (derived) from the device identity and the user-typed string according to ONVIF 1.0 specification.
Onvif 2.0+	The password is computed (derived) from the device identity and the user-typed string according to ONVIF 2.0 (or later) specification.

10. Storage Page

The Storage page allows the user to mount/unmount USB storage media, enable/disable/start/stop recording and list/preview stored media files.

The Storage page displays three panels:

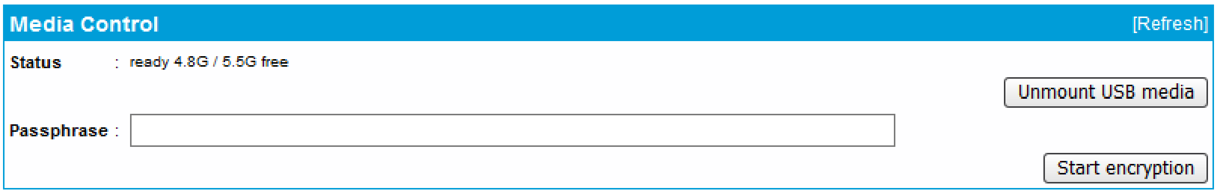
- Media Control
- Recording Control
- Stored Media

Encrypted files

- When you download a `secure-*` file, the Picolo.net HD1 always returns the AES-encrypted contents as found on the USB media.
- When you preview a `secure-*` file in the media preview page, the Picolo.net HD1 decrypts the file on the fly and displays it in your web browser only if the passphrase used to encrypt it is currently known to the device.

Media Control Panel

"Connecting an External USB Drive" on page 96



Media Control

Status : ready 4.8G / 5.5G free

Passphrase :

Media Control panel

The Media Control panel allows to control the USB media.

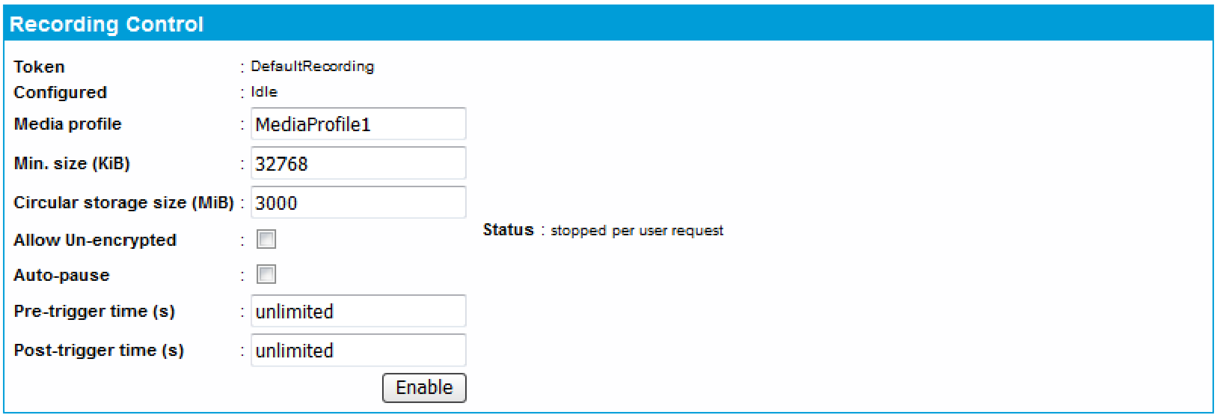
- Picolo.net HD1 automatically detects USB mass storage devices as they are plugged in.
  - □ Devices with a FAT32, exFAT and EXT4 file system are automatically mounted when plugged in.
  - □ If the device is not listed, press [Refresh] on the media control panel to update the page.
- When mounted, the status field turns reports a ready condition together with the remaining and the total capacity of the USB media.
- If Picolo.net HD1 does not recognize the file system format or if the root / partition information is corrupted on the device:
  - □ It reports a "bad disk" status.

- □ It offers to format the first (or only) partition of the device in EXT4.

You can safely unplug a device with the "bad disk" status.

- To unmount a USB media:
  - □ Click on the unmount USB media button.
  - □ The recording job stops and all pending data are written.
  - □ The status turns to 'idle safe for removal' to confirm the operation.
  - □ Remove the device or manually mount the partition again by clicking the mount USB media button.

Recording Control Panel



Recording Control

Token	: DefaultRecording
Configured	: Idle
Media profile	: MediaProfile1
Min. size (KiB)	: 32768
Circular storage size (MiB)	: 3000
Allow Un-encrypted	: □
Auto-pause	: □
Pre-trigger time (s)	: unlimited
Post-trigger time (s)	: unlimited

Status : stopped per user request

Recording Control panel

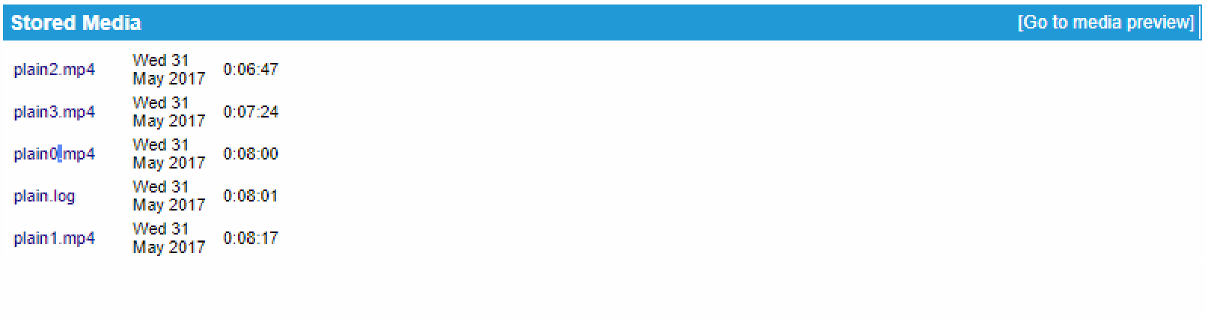
In the Recording Control panel, you can:

- Enable and disable the recording.
- Start, stop and resume the recording.
- Trigger the recording using the GPIO Alarm signal.

The recording enable/disable settings is persistent. It is not affected by device reboot or by a power on/off/on cycle.

See also "Managing the Recording" on page 80 for detailed procedures.

Stored Media Panel



Stored Media [Go to media preview]

plain2.mp4	Wed 31 May 2017	0:06:47
plain3.mp4	Wed 31 May 2017	0:07:24
plain0.mp4	Wed 31 May 2017	0:08:00
plain.log	Wed 31 May 2017	0:08:01
plain1.mp4	Wed 31 May 2017	0:08:17

Stored Media panel

The Stored Media panel lists the files recorded by the Picolo.net encoder on the USB media.

To download a file, click on the file name.

11. Layers Page

The Layers page allows the user to configure and position the OSD (On Screen Display) content.

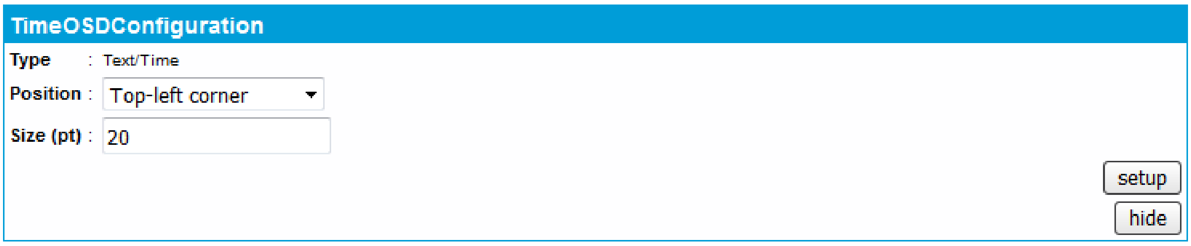
The Storage page displays three panels:

- TimeOSDConfiguration
- UserOSDConfiguration (you can use any Chinese, Cyrillic, Greek, Japanese, Korean and Latin characters).
- AutoOSDConfiguration

UserOSDConfiguration and AutoOSDConfiguration are not rendered on the HDMI output, but only on the encoded bitstreams.

The TimeOSDConfiguration is rendered on both encoded bitstreams and HDMI output.

TimeOSDConfiguration Panel



TimeOSDConfiguration

Type : Text/Time

Position : Top-left corner

Size (pt) : 20

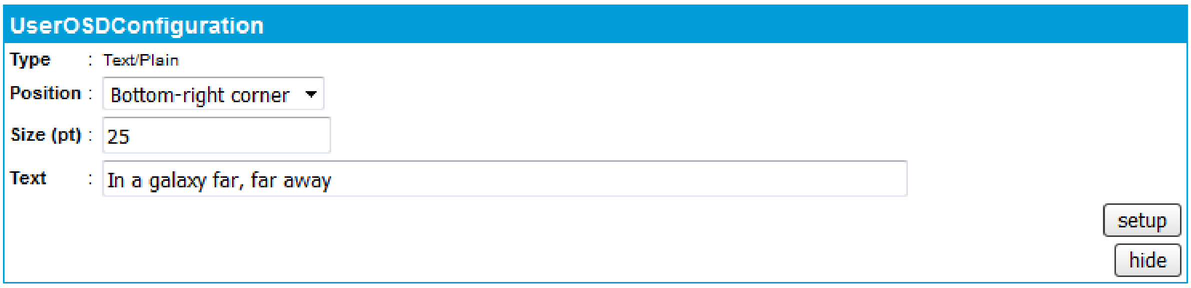
TimeOSDConfiguration panel

The TimeOSDConfiguration panel controls a hardware time stamp incremented at every frame.

Use the following controls to define the text display:

- □ Select the Position of the text.
- □ Set the Size (pt) of the font used.
- □ Click on the setup button to apply the changes if the layer is shown.
- □ Click on the hide / show button to enable or disable the text display.

UserOSDConfiguration Panel



UserOSDConfiguration

Type : Text/Plain

Position : Bottom-right corner

Size (pt) : 25

Text : In a galaxy far, far away

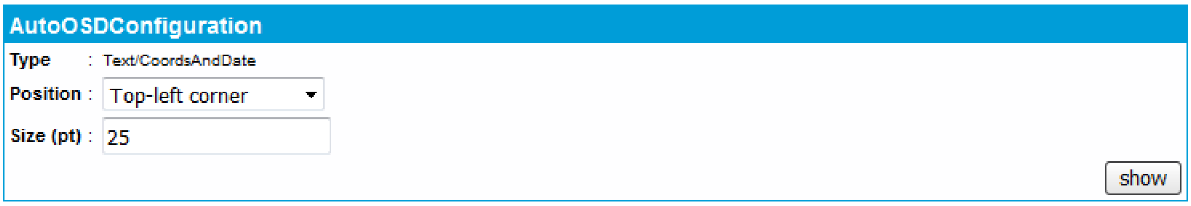
UserOSDConfiguration panel

The UserOSDConfiguration panel controls a hardware time stamp incremented at every frame.

Use the following controls to define the text display:

- Select the Position of the text.
- Set the Size (pt) of the font used.
- Enter the Text you want to display on the screen.
- Click on the setup button to apply the changes if the layer is shown.
- Click on the hide / show button to enable or disable the text display.

AutoOSDConfiguration Panel



AutoOSDConfiguration

Type : Text/CoordsAndDate

Position : Top-left corner

Size (pt) : 25

AutoOSDConfiguration panel

The AutoOSDConfiguration panel controls a hardware time stamp incremented at every frame.

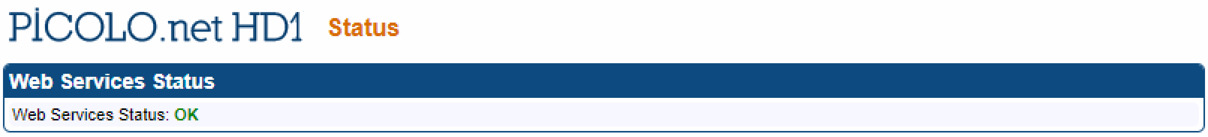
Use the following controls to define the text display:

- Select the Position of the text.
- Set the Size (pt) of the font used.
- Click on the setup button to apply the changes if the layer is shown.
- Click on the hide / show button to enable or disable the text display.

12. Hidden Pages

12.1. Check Status Page

The Check Status page URL is: `http://[device-ip-address]/check-status` for instance : `http://192.168.12.217/check-status`.



Web Services Status

Web Services Status: OK

Web Services Status panel

The Web Services Status field OK indicates that all the web services are up.

PART II

USING THE SERVICES

1. Using the Device Service

1.1. The Device Functions

See also: The ONVIF Device Service | The Proprietary Device Service in the Reference manual

Retrieve the device temperature (proprietary)

Use the `GetTemperature` function to read the device internal temperature:

- The request message `GetTemperatureRequest` has no content.
- The response message `GetTemperatureResponse` contains (all values are in °C):
  - □ The CPU in-chip temperature (`DieCpu`), updated every minute, in an XML data structure of the `float` type.
  - □ The video processing in-chip temperature (`DieVideo`) in an XML data structure of the `float` type.
  - □ The box temperature (`BoardAmbient`) in an XML data structure of the `float` type.
  - □ A human-readable summary of all this information in `Temperature` in an XML data structure of the `string` type.

Retrieve the users lockout configuration (proprietary)

Use the `GetLocksConfiguration` function to read the current configuration for users lockout on failed logins:

- The request message `GetLocksConfigurationRequest` has no content.
- The response message `GetLocksConfigurationResponse` contains a `LocksConfiguration` with the following fields:
  - □ The amount of bad passwords tolerated (`retries`) before a user account gets locked, in an XML data structure of the `int` type.
  - □ The `duration` (in seconds) of user accounts lockout, in an XML data structure of the `int` type.

Configure the users lockout configuration (proprietary)

Use the SetLocksConfiguration function to modify the users lockout configuration:

- The request message SetLocksConfigurationRequest contains a LocksConfiguration with the following fields:
  - □ The amount of bad passwords tolerated (retries) before a user account gets locked, in an XML data structure of the int type.
  - □ The duration (in seconds) of user accounts lockout, in an XML data structure of the int type.
- The response message SetLocksConfigurationresponse has no content.

Retrieve the device date and time (vendor-specific extensions)

Use the GetSystemDateAndTime function to read the device date and time:

- The request message GetSystemDateAndTimeRequest has no content.
- The response message GetSystemDateAndTimeResponse contains:
  - □ The date and the time of the device in an XML data structure of the SystemDateTime type (following the ONVIF standard).
  - □ Additional vendor-specific data in a structure of the SystemDateTimeExtension type.

Configure the device date and time (vendor-specific extensions)

Use the SetSystemDateAndTime function to set the device date and time:

- The request message SetSystemDateAndTimeRequest contains:
  - □ The date and the time of the device in an XML data structure of the SystemDateTime type (following the ONVIF standard).
  - □ Additional vendor-specific data in a structure of the SystemDateTimeExtension type.
- The response message SetSystemDateAndTimeResponse has no content.

1.2. The Device Types

The SystemDateTimeExtension type

An ONVIF-defined extension of the SystemDateTime type is composed of the following vendor-defined fields:

- □ An optional <GpsEnable> element of the boolean type that configures if the Picolo.net HD1 is allowed to use the attached USB GPS as an NTP time source.
- □ An optional, read-only <GpsAvailable> element of the boolean type that reports whether a USB GPS is attached to Picolo.net HD1.

□ An optional `<TimeZoneString>` element that captures a user-friendly name for the defined time zone.

When one of these fields is missing during a `SetSystemDateAndTime` call, the existing configuration is preserved.

This extension is defined in the onvif.xsd file.

2. Using the Recording Service

2.1. The Recording Processing Chain

See also: The ONVIF Recording Service | The Proprietary Recording Service in the Reference manual

Use the functions of the recording service to manage your network storage devices and your cameras with embedded storage:

- □ Configure the local media profile you want to record (on attached USB 2 media).
- □ Start and stop the recording job.
- □ Manage the storage content and status.

The recording specifications

- Supported file systems: FAT32, exFAT, EXT4
- Cryptographic options: eCryptfs
- Supported partition tables: MBR, GPT

The recording processing chain

The recording processing chain is composed of the following elements:

- □ 1 stream multiplexer
- □ 1 optional AES encryption layer
- □ 1 retention manager

The stream multiplexer

The stream multiplexer pulls one of the H.264 streams from the video processing chain and optionally the AAC stream from the audio processing chain and combines them to get one streaming-ready ISO IEC 14496 (MP4) container.

Use a Media Profile token listed in the ONVIF `RecordingConfiguration` entry to select the audio and the video streams.

The files generated by the stream multiplexor always consist of an integer number of "Group of Pictures" sets. Each set is made of 1 IDR frame and 0 or more P frames. You can use a proprietary extension to the ONVIF Recording Job element to define how big a file must grow before allowing the multiplexor to start a new file.

The file names follow the pattern `<protected?>-<session-identifier>--<sequence-number>-<hour>-<minutes>-<seconds>.mp4`, where:

- `<protected?>` is either `plain` (no encryption) or `secure` (128-bit AES).
- Every resuming of the recording process generates a new (numerical) `session-identifier`.

If other sessions are already present on the USB media, the new session identifier is greater than the previously existing session identifiers.

- The sequence numbers are assigned every time a new file is required.

The AES encryption layer

If enabled, the AES layer transparently encrypts and decrypts the files as they are written to / read from the USB media. It means that the disk only contains encrypted files while the processes running on the Picolo.net HD1 see them as normal files. The appropriate 128-bit AES keys are derived from passphrases. The AES layer automatically detects if an existing file on the USB media is encrypted with a given passphrase and attempts decryption only when the passphrase fingerprint computed internally and the one stored in the file match.

When only the AES layer is used, the AES layer is enabled temporarily (until the device reboots) through proprietary extensions to the ONVIF Recording service.

When the AES layer is combined with the OpenPGP key(s), the AES layer is permanently enabled. A random passphrase is generated at every boot and the AES layer is automatically enabled. The passphrase is then encrypted with the OpenPGP key(s) uploaded on the Picolo.net HD1 for later decryption.

The retention manager

The retention manager monitors the files produced by the recording chain and decides when to delete which file in order to guarantee that a desired amount of video is available on the USB media.

2.2. The Recording Functions - Configuration and Contents

Get the extended recording configuration (proprietary)

Use the `GetExtendedRecordingConfiguration` function to retrieve additional proprietary information to complete a response item to ONVIF `GetRecordings` call.

- The request message `GetExtendedRecordingConfigurationRequest` contains:
  - □ A `ConfigurationToken` that identifies the configuration to retrieve in an XML string of the `RecordingReference` type.
- The response message `GetExtendedRecordingConfigurationResponse` contains:
  - □ A `Configuration` in an XML data structure of the `ExtendedRecordingConfiguration` type.

Set the extended recording configuration (proprietary)

Use the `SetExtendedRecordingConfiguration` function to configure the proprietary extension of a recording.

- The request message `SetExtendedRecordingConfigurationRequest` contains:
  - □ A `ConfigurationToken` that identifies the configuration to retrieve in an XML string of the `RecordingReference` type.
  - □ A `Configuration` in an XML data structure of the `ExtendedRecordingConfiguration` type.
- The response message S`etExtendedRecordingConfigurationResponse` has no content.

Count the storage contents (proprietary)

Use the `CountStorageContents` function to:

- □ Scan the files produced by the recording chain.
- □ Report how many of them have been created during a specified time range.

- The request message `CountStorageContentsRequest` contains:
  - □ An optional `From` lower limit for the time range in an XML structure of the `xs:dateTime` type.
  - □ An optional `Until` upper limit for the time range in an XML structure of the `xs:dateTime` type.

- The response message CountStorageContentsResponse contains:
    - □ The number of Items matching the request in an XML structure of the int type.
    - □ The oldest and the latest time stamps in the defined time range (From and Until, respectively) in an XML structures of the xs:dateTime type.

List the storage contents (proprietary)

Use the ListStorageContents function to report the files produced by the recording chain over a specified time range.

- The request message ListStorageContentsRequest contains:
    - □ An optional From lower limit for the time range in an XML structure of the xs:dateTime type.
    - □ An optional Until upper limit for the time range in an XML structure of the xs:dateTime type.
    - □ An optional MaxItems that restricts the number of clips included in the reply.
    - □ An optional Continuation token to request the next batch of files when the response contains this same token.
- The response message ListStorageContentsResponse contains:
    - □ A collection of Clips, each describing one file in an XML structure of the StoredMediaClip type.
    - □ A Continuation token if the directory contains more than MaxItems files. Use this token in your request to retrieve the next batch of files contained in the directory.

2.3. The Recording Functions - AES

Get the AES storage status (proprietary)

Use this function to check if the AES cryptography layer is currently applied to the recording chain.

- The request message GetAESStorageStatusRequest has no content.
- The response message GetAESStorageStatusResponse contains:
    - □ The name of the Directory on which the cryptography layer is or may be applied in an XML data structure of the string type.
    - □ The current status of encryption in an XML data structure of the EncryptionEnum type:
None if cryptography is disabled.
AES128 if 128-bit AES is enabled.

Unlock the AES storage (proprietary)

Use this function to:

- □ Activate the AES cryptography layer in the recording chain.
- □ Unlock the files from previous sessions encrypted with the provided passphrase.
- □ Enable the encryption of incoming files with the provided passphrase.

- The request message `UnlockAESStorageRequest` contains:
  - □ The `PassPhrase` used to produce the 128-bit AES key, converted to base-64 using the conventions of `xs:base64Binary` in an XML data structure of the `string` type.
  - □ The `Directory` on which you want to apply the cryptography layer, as returned by `GetAESStorageStatus`.
- The response message `UnlockAESStorageResponse` has no content.

This call is required only when you have no OpenPGP key installed. The unlocking is temporary and you need to repeat it when the disk is re-inserted or when the device is rebooted. When OpenPGP is installed, it will automatically activate the AES layer for each new session.

The passphrase to key conversion is a deliberately long process (several minutes).

Lock the AES storage (proprietary)

Use this function to:

- □ Deactivate the AES cryptography layer in the recording chain.
- □ Lock again the encrypted files.

The passphrase entered in `UnlockAESStorageRequest` and the corresponding key are purged from memory.

- The request message `LockAESStorageRequest` has no content.
- The response message `LockAESStorageResponse` has no content.

2.4. The Recording Functions - USB

Mount a USB storage (proprietary)

Use this function to:

- □ Allow your Picolo.net HD1 to access the file system on a USB media (mounting).
- □ Prepare your Picolo.net HD1 for the media removal (un-mounting).

- The request message `MountUSBStorageRequest` contains:
  - □ The `Umount` flag that indicates if you want to un-mount the media in an XML data structure of the `boolean` type.

  When a Picolo.net HD1 receives an `Umount` request, it:

  - □ Stops the current recording job, if any.
  - □ Flushes the pending writes to the disk.
  - □ Prevents any further I/O access on the disk.
  - □ Sends a reply.

  When you receive the reply, you may safely remove the disk.

- The response message `MountUSBStorageResponse` has no content.

  When a Picolo.net HD1 receives a "mount" request, it:

  - □ Checks the presence of a USB disk.
  - □ Gives the recording service and the web pages access to the file system.

Resume the USB storage operation (proprietary)

When you connect or reconnect a USB media to a Picolo.net HD1 (mounting), the file system is automatically available but the recording job stays in the suspended state.

Use this function to resume the recording job and automatically test its state:

- The request message `ResumeUSBStorageRequest` has no content but returns a SOAP fault if any of the following conditions occurs:
  - □ There is no mass media storage on the USB ports.
  - □ The media is not ready yet.
  - □ The recording job does not resume properly.
  - □ There is no configured recording job.

  `ResumeUSBStorageRequest` fails with a `ter:DiskNotReady` sub-code.

- The response message `ResumeUSBStorageResponse` has no content.

  It is sent if and only if either:

  - □ The recording successfully restarted.
  - □ The recording is already running.

2.5. The Recording Types

The StoredMediaClip type

A structure type defining one stored file, composed of:

- An element <Name> of the string type that gives the name of the file.
- An element <Uri> of the anyUri type that gives the link to download or preview the file.
- An element <Start> of the xs:dateTime type that reports the creation time of the file. It can be used to approximate the time stamp of the first frame in the file.
- An element <End> of the xs:dateTime type that reports the closing time of the file. It can be used to approximate the time stamp of the last frame in the file.

The ExtendedRecordingConfiguration type

An extension of tokenHolder and a companion type to the ONVIF GetRecordingsResponseItems composed of:

- An element <FileSize> of the KibiBytes type that configures the minimum size that a recording file should reach before the recording may decide to create a new file.

Each file always contains an integer number of Group-of-Pictures, whatever the value of this parameter.

- An element <Guaranteed> of the MebiBytes type that configures the minimum amount of video (in MiB) for which we should guarantee that files won't be recycled.
- An optional element PauseOnSignalLost of the boolean type that triggers conditional recording based on the video signal status monitoring.
- An optional element OnEvent of the TriggeredRecordingConfiguration type that configures GPIO-triggered recording.
- An element <AllowUnencrypted> of the boolean type that enables the job recording if no encryption is active (false by default).

The TriggeredRecordingConfiguration type

A structure composed of:

- An element <PreTrigger> of the duration type that defines the minimum recording duration to be preserved before the event occurred.
- An element <PostTrigger> of the duration type that defines the minimum recording duration to be preserved after the event occurred.

3. Using the Media Service

3.1. The Media Functions

See also: The ONVIF Media Service | The Proprietary Media Service in the Reference manual

Trigger the auto setup

Use this function to trigger "The Auto Setup Profiles" on page 72.

- The request message `AutoSetupRequest` has no content.
- The response message `AutoSetupResponse` has no content.

Get the Picolo audio outputs

Use this function to enumerate the audio output ports available in the device.

- The request message `GetPicoloAudioOutputsRequest` has no content.
- The response message `GetPicoloAudioOutputs` contains:
  - Zero or more `PicoloAudioOutputs` elements of the `PicoloAudioOutput` type: one per audio output port available in the device.

Get the Picolo audio output configuration

Use this function to retrieve the configuration of an audio output port.

- The request message `GetPicoloAudioOutputConfigurationRequest` contains:
  - The token name of the audio output port in an XML data structure of the `string` type.
- The response message `GetPicoloAudioOutputConfigurationResponse` contains:
  - The configuration of the audio output port in an XML data structure of the `PicoloAudioOutputConfiguration` type.

Configure the Picolo audio output

Use this function to configure an audio output port.

- The request message `SetPicoloAudioOutputConfigurationRequest` contains:
    - □ The token name of the audio output port in an XML data structure of the `string` type.
    - □ The configuration of the audio output port in an XML data structure of the `PicoloAudioOutputConfiguration` type.
- The response message `SetPicoloAudioOutputConfigurationResponse` has no content.

The configurations are persistent. The audio output ports reconnect automatically during the boot of the device.

Retrieve the URI

Use this function to determine the URI you should use to retrieve the live captured media over the TLS-protected connections.

The data stream carried over these HTTPS connections can be either MP4 file (containing H.264 video + AAC audio) or RTSP-tunnelled-over-HTTP. It is determined by the `format` argument passed to the service. When you have retrieved the URI, you must set up the proper protocol stack to communicate with the server located by the URI.

- The request message `GetPicoloHttpsUriRequest` contains:
    - □ The token name of a Media Profile in an XML string of the `ReferenceToken` type.
    - □ The desired MIME type transported in an XML data structure of the `PicoloHttpsStreamMime` type.
- The response message `GetPicoloHttpsUriResponse` contains:
    - □ The Uniform Resource Identifier (URI) to access the stream in an XML data structure of the `MediaURI` type.

Force the input reconfiguration

The Picolo.net HD1 stores the resolution and the frame rate of the video source in the `VideoSource` element of its configuration. The Picolo.net HD1 uses this configuration to decide which video signal is valid and which on is invalid.

Use this function to override the current settings of the `VideoSource` with the characteristics of the current signal.

- The request message `ForceInputConfigurationRequest` contains:
  - □ An optional `UseHDMI` element of the `boolean` type that sets the reconfiguration to use the HDMI signal (`false` by default).
  - □ An optional `UseSDI` element of the `boolean` type that sets the reconfiguration to use the SDI signal (`false` by default).
  - □ A request must have one element to be valid.
- The response message `ForceInputConfigurationResponse` has no content.

If you modify the `VideoSource` configuration to match the signal on a connector, it does not force the video pipeline to switch to the corresponding connector.

The scaling and the framerate decimation parameters defined in `VideoEncoderConfiguration` elements still apply.

3.2. The Media Types

The `PicoloAudioOutput` type

An extension of the `DeviceEntity` type, a base class for physical entities like inputs and outputs.

The element attribute @token contains the token name, a unique identifier referencing the audio output.

The `PicoloAudioOutputConfiguration` type

An extension of the `ConfigurationEntity` type composed of:

- □ An element `<SourceURI>` of the `anyURI` type.
- □ An optional element `<UserName>` of the `string` type.
- □ An optional element `<Password>` of the `string` type.

- The `<SourceURI>` element contains the URI of an RTSP audio stream. An empty `<SourceURI>` disables a currently configured `PicoloAudioOutput`.
- The `<UserName>` and `<Password>` elements contain the credentials for authentication on the RTSP server.

The `PicoloHttpsStreamMime` type

An enumeration type defining the MIME type that you want to retrieve over HTTPS.

The supported values are:

- □ `video` / `mp4`
- □ `application` / `x-rtsp-tunnelled`

- You can pass `video` / `mp4` format as-is to an HTML5 compliant web browser for playback.
- You can use the `x-rtsp-tunnelled` format in RTSP-based libraries such as gstreamer 1.0.

The `PicoloAudioOutput` event message

This event reports the changes of state related to the audio outputs:

- □ An invalid user name or password for the RTSP authentication.
- □ The stream issues.
- □ The network issues.

3.3. The Auto Setup Profiles

Picolo.net HD1 implements a procedure called "Auto Setup Profiles" both in the proprietary API and in the device web pages that:

- □ Erases all existing ONVIF media profiles.
- □ Creates 1 ONVIF media profile for each currently connected camera.

It is executed:

- □ When the user requires it:
  - By pressing the corresponding button in the "Media Profiles Page" on page 13 web page,
  - Or by calling the API function.
- □ At boot time, for the cameras that do not have a workable ONVIF media profile.

The generated ONVIF media profiles bind the corresponding video source object to a particular combination of:

- □ A video source configuration,
- □ A video encoder configuration,
- □ PTZ configuration objects.

Euresys reserves the rights to modify the composition of the collection and/or the settings of the configuration objects in future firmware upgrades.

4. Using the Event Service

4.1. The Event Functions

See also: The ONVIF Event Service in the Reference manual

Topic "VideoSource" (ONVIF)

Sub-topic "VideoSource/Signal" (proprietary)

This event reports the connectivity changes on the effective connector of a video source:

- □ It contains a `VideoSourceToken` that identifies the subject source.
- □ It contains a `Standard` string that reports whether signal was lost (`NoSignal`) or locked (`<number-of-lines>p<framerate>`).

Sub-topic "VideoSource/ConnectorSignal" (proprietary)

This event reports the connectivity changes on any connector that a video source can use.

- □ It contains a `VideoSourceToken` that identifies the subject source.
- □ It contains a `Standard` string that reports whether signal was lost (`NoSignal`) or locked (`<number-of-lines>p<framerate>`).

Topic "Device" (ONVIF)

Sub-topic "Device/DigitalInput" (proprietary)

This event reports the state changes on an digital input (alarm) connector:

- □ It contains a `DigitalInputToken` that identifies the subject connector.
- □ It contains a `State` string that reports whether the input is currently driven high or low by the external circuit.

Sub-topic "Device/SystemButton" (proprietary)

This event reports the actions on the control buttons of the device:

- □ It contains a `SystemButtonToken` that identifies the subject button.
- □ It contains `FactorySettings` for the 'revert to factory settings' button
- □ It contains a `State` string that reports whether the button is currently pressed or released.

5. Using the Device IO Service

5.1. The Device IO Functions

See also: The ONVIF Device IO Service | The Proprietary Device IO Service in the Reference manual

See section "Sending Custom Commands over the Serial Port" on page 98 for additional information on custom commands.

Configure an alarm input port

Use the `SetDigitalInputConfiguration` function to configure an alarm input port.

- The request message `SetDigitalInputConfigurationRequest` contains:
  - □ The token name of the alarm input port in an XML data structure of the `string` type.
  - □ The configuration of the alarm input in an XML data structure of the `InputConfiguration` type.
- The response message `SetDigitalInputConfigurationResponse` has no content.

Retrieve the configuration of an alarm input port

Use the `GetDigitalInputConfiguration` function to retrieve the configuration of an alarm input port.

- The request message `GetDigitalInputConfigurationRequest` contains:
  - □ The token name of the alarm input port in an XML data structure of the `string` type.
- The response message `GetDigitalInputConfigurationResponse` contains:
  - □ The configuration of the alarm input in an XML data structure of the `InputConfiguration` type.

Retrieve the state of an alarm input port

Use the `GetDigitalInputState` function to retrieve the state of an alarm input port.

- The request message `GetDigitalInputStateRequest` contains:
  - □ The index of the alarm input port.
  - □ The token name of the alarm input port in an XML data structure of the `string` type.
- The response message `GetDigitalInputStateResponse` contains:
  - □ The state of the alarm inputs in an XML data structure of the `State` type.

Retrieve the serial port configuration (vendor-specific extensions)

Use the `GetSerialPortConfiguration` function to read the serial port configuration:

- The request message `GetSerialPortConfigurationRequest` has no content.
- The response message `GetSerialPortConfigurationResponse` contains:
  - □ The configuration of the serial port in an XML data structure of the `SerialPortConfiguration` type (following the ONVIF standard).
  - □ Additional vendor-specific data in a structure of the `SerialPortConfigurationExtension` type.

Configure the serial port (vendor-specific extensions)

Use the `SetSerialPortConfiguration` function to configure the serial port:

- The request message `SetSerialPortConfigurationRequest` contains:
  - □ The configuration of the serial port in an XML data structure of the `SerialPortConfiguration` type (following the ONVIF standard).
  - □ Additional vendor-specific data in a structure of the `SerialPortConfigurationExtension` type.
- The response message `SetSerialPortConfigurationResponse` has no content.

5.2. The Device IO Types

The `State` type

This type is composed of:

- □ An element `<State>` of the `eur:InputStateEnum` type.

- The element `<State>` specifies the state of the alarm input port.
  The possible values are:
  - □ **`OPEN`**: the alarm input port measures a high-impedance that represents an open contact or an unused port.
  - □ **`HIGH`**: the alarm input port measures a voltage above the voltage threshold.
  - □ **`LOW`**: the alarm input port measures a voltage below the voltage threshold that represents a closed contact or a logical low level.

The `InputConfiguration` type

This type is composed of:

- □ An element `<VoltageThreshold>` of the `eur:VoltageThresholdEnum` type.
- □ An element `<TimingFilter>` of the `eur:TimingFilterEnum` type.
- □ An element `<EnableEvents>` of the `xs:boolean` type.

- ● The element `<VoltageThreshold>` specifies the voltage threshold of the alarm input port.
  The possible values are:
  - □ **`TTL`**: the threshold voltage is 1.4 V and is adapted to TTL devices, 3 V CMOS devices or potential-free contacts.
  - □ **`5V CMOS`**: the threshold voltage is 2.5 V and is adapted to 5 V CMOS devices.
  - □ **`12V`**: the threshold voltage is 6 V and is adapted to 12 V or higher CMOS devices.
- ● The element `<TimingFilter>` specifies the strength (time constant) of the noise filter of the alarm input port.
  The possible values are:
  - □ **`OFF`**: the noise filter is set to the minimal strength.
  - □ **`10ms`**: the noise filter is set to the medium strength and filters out signal transients shorter than 10 milliseconds.
  - □ **`100ms`**: the noise filter is set to the maximal strength and filters out signal transients shorter than 100 milliseconds.

The `SerialPortConfigurationExtension` type

An ONVIF-defined extension of the `SerialPortConfiguration` type is composed of the following vendor-defined fields:

- □ An optional `<ScanVisca>` element of the `boolean` type that configures if the Picolo.net HD1 sends VISCA discovery messages on the corresponding serial port.

If this field is missing during a `SetSerialPortConfiguration` call, the existing configuration is preserved.

This extension is defined in the deviceio.wsdl file.

5.3. Using the PTZ Service

The PTZ Functions

See also: The ONVIF PTZ Service | The Proprietary PTZ Service in the Reference manual

See section "Sending Custom Commands over the Serial Port" on page 98 for additional information on custom commands.

Configure a node (proprietary)

Use the `SetPelcoNodeAddressConfiguration` function to configure a specific PTZ node:

- The request message `SetPelcoNodeAddressConfigurationRequest` contains:
  - □ The configuration of the PTZ node in a XML data structure of the `eur:PelcoNodeAddressConfiguration` type.
- The response message `SetPelcoNodeAddressConfigurationResponse` has no content.

Read the configuration of a node (proprietary)

Use the `GetPelcoNodeAddressConfiguration` function to read the configuration of a specific PTZ node:

- The request message `GetPelcoNodeAddressConfigurationRequest` contains:
  - □ The token of the PTZ node configuration in a XML data structure of the `string` type.
- The response message `GetPelcoNodeAddressConfigurationResponse` contains:
  - □ The configuration of the PTZ node in a XML data structure of the `eur:PelcoNodeAddressConfiguration` type.

Read the configuration of all the nodes (proprietary)

Use the `GetPelcoNodeAddressConfigurations` function to read the configurations of all PTZ node:

- The request message `GetPelcoNodeAddressConfigurationsRequest` has no content.
- The response message `GetPelcoNodeAddressConfigurationsResponse` contains:
  - □ The configurations of the PTZ nodes, each in a XML data structure of the `eur:PelcoNodeAddressConfiguration` type.

ERSP - Read the URI of a serial port

Use the GetTransparentSerialURI function to start an instance of the **ERSP** (**Euresys Remote Serial Protocol**) and read its URI (Uniform Resource Identifier):

- The request message GetTransparentSerialURIRequest contains:
  - □ The serial port token to bind to the **ERSP** server instance in an XML data structure of the ReferenceToken type.

If an instance is already running, this operation fails and returns a ter:Action SOAP fault.

- The response message GetTransparentSerialURIRequest contains:
  - □ The URI you must use to connect to the newly started server with the format: ersp://<device-name-or-address>:<port-number>/<one-time-authentication-key>

A running ERSP server gets exclusive access to the corresponding serial port identified. Further PTZ, imaging or SendReceiveSerialCommand calls to the serial port fail and return a ter:Action SOAP fault.

ERSP - Stop and release a serial port

This operation stops an instance of the **ERSP** (**Euresys Remote Serial Protocol**) and resumes access to the serial port for other software components:

- The request message StopTransparentSerialRequest contains:
  - □ The serial port token of the instance you want to terminate in an XML data structure of the ReferenceToken type.
- The response message StopTransparentSerialResponse has no content.

The PTZ Types

The PelcoNodeAddressConfiguration type

This complex type is composed of:

- □ A root element <Configuration>.
- □ A root element attribute @token. It is the unique identifier of the PTZ node configuration.
- □ A child element <Address> of the xs:unsignedByte type. It contains the physical address of the PTZ node in the range [0:255].
- □ An optional child element MinFocusSpeed of the xs:unsignedByte type that contains a PELCO speed limit in the range [0:3].
- □ An optional child element MaxFocusSpeed of the xs:unsignedByte type that contains a PELCO speed limit in the range [0:3].

PART III

PROCEDURES AND USE CASES

1. Managing the Recording

See also: "Storage Page" on page 52 | "Connecting an External USB Drive" on page 96

Use Recording Control panel in the Storage page to manage your recording settings.

Enabling / Disabling the recording

Enable

1. Ensure that the USB media is correctly mounted.
2. Ensure that the selected media profile exists and uses codecs compatible with MP4 file format (H.264 and AAC only).
3. If required, adjust the Min. size parameter in the "Recording Control Panel" on page 53 to set minimum file size.

   Based on Min. size parameter, a new file is created every time a new GOP starts and the file is larger than this size.
4. Configure the amount of storage dedicated to the Picolo.net HD1 on the device if circular recording is used or use Unlimited in the Circular storage size.
5. Ensure that the encryption is correctly configured and complete or that the recording of unencrypted files is enabled.
6. Click on the Enable button.

   This automatically starts the recording if a disk is connected.

The recording enable / disable settings is persistent. It is not affected by device reboot or by a power on / off / on cycle.

Disable

- Click on the Disable button.

The recording enable / disable settings is persistent. It is not affected by device reboot or by a power on / off / on cycle.

Starting / Stopping the recording

Start

- The recording automatically starts when you enable the recording job.

Stop

- Click on the Stop button.

Resume

1. Ensure that the USB media is correctly mounted.
2. Ensure that the recording is enabled.
3. Click on the Start button.

Triggering the recording

You can control the beginning of the recording based on an external event detection, using the GPIO Alarm connector on your Picolo.net HD1:

- The Picolo.net HD1 considers both the rising and the falling edges of the GPIO Alarm as individual triggers.
- Configure the recording duration before and after the triggering event in the Recording Control web page or with the `SetExtendedRecordingConfiguration` of HD1 proprietary web services.
- To activate the triggering, enter a value (in seconds) in both the Pre-trigger time and the Post-trigger time fields.
- To disable the triggering, empty both the Pre-trigger time and the Post-trigger time fields.
- The Min. size field (`FileSize` parameter) still defines the start and the length of the file.

When you enable the triggering, the Picolo.net HD1 only keeps the files that cover at least one event.

2. Using Encryption

2.1. Using the OpenPGP and AES Encryption

OpenPGP and AES

A mixed **OpenPGP** and **AES** encryption chain is applied to the media files stored on a USB drive:

- The **AES** layer allows the encryption and decryption of media files. The **AES** keys are derived from a passphrase generated at every boot.
- The **OpenPGP** layer allows the encryption and decryption of the **AES** keys. The **OpenPGP** encryption standard is based on pairs of private and public keys. The two matching private and public keys revert each other actions.

With the **OpenPGP+AES** encryption used in Picolo.net HD1, the protection mechanism works as follows:

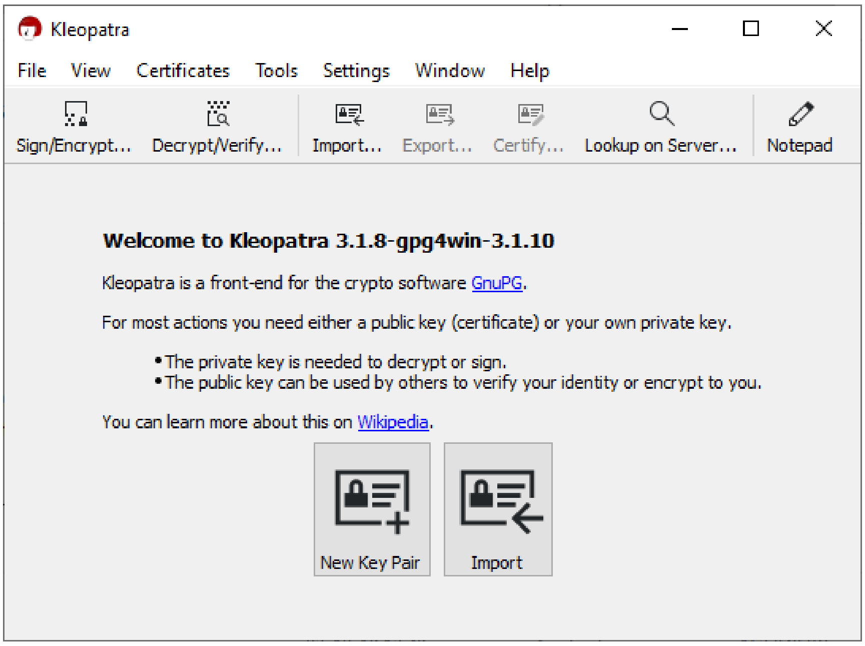
1. Each time the Picolo.net HD1 boots, it creates a new **AES** key (passphrase) and stores it in its volatile memory.
2. You upload the **OpenPGP** public key on the non-volatile memory of the Picolo.net HD1 device.
3. When the Picolo.net HD1 records media files and stores them on the USB drive, it secures them with the **AES** encryption.
4. The current **AES** passphrase, encrypted with the **OpenPGP** public key installed, is copied in a separate folder of the USB drive.
5. When users get the USB drive, they use the **OpenPGP** private key received to unseal the files. The appropriate **AES** keys are then automatically used to decrypt the media files.

NOTE
The software programs referred to in this section are solely intended to explain how to operate Picolo.net HD1.

We do not endorse these programs. Other programs complying to IETF RFC4880 allow you to perform the same tasks.

Create a pair of keys with Kleopatra from GPG4Win

1. Download and install Gpg4Win.
2. Click the New Key Pair button to enter the Key Pair Creation Wizard.


Kleopatra
File View Certificates Tools Settings Window Help
Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
New Key Pair
Import

Welcome to Kleopatra 3.1.8-gpg4win-3.1.10

Kleopatra is a front-end for the crypto software GnuPG.

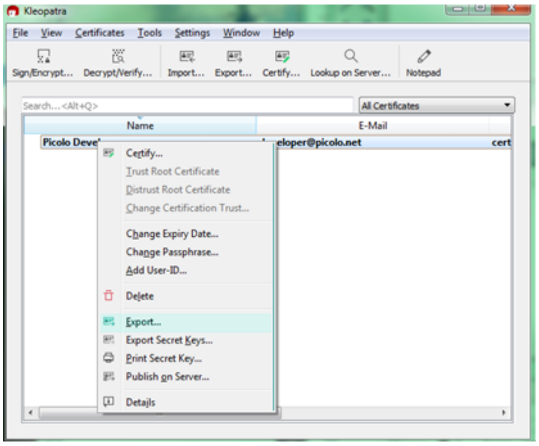
For most actions you need either a public key (certificate) or your own private key.

- The private key is needed to decrypt or sign.
- The public key can be used by others to verify your identity or encrypt to you.

You can learn more about this on Wikipedia.

3. Enter the name and the e-mail address identifying the owner of the new keys
4. Click Advanced Settings and edit the following settings if requested:
   - □ the certificate validity in the Valid until field
   - □ the encryption algorithm in Key Material
     We tested the key pair generation with the default settings (RSA 2048 bits).
5. When requested, enter twice the passphrase to be able to create the key.

   The passphrase is used only on your local computer to keep the private key safe on your disk.
6. When the key pair is created, right-click on your key and select Export in the contextual menu.



The software creates a new file with the public key.

The Picolo.net HD1 server preferably uses e-mail addresses to identify key owners.
You can identify a key without an associated e-mail address only through its hexadecimal signature.

Create a pair of keys with the GPG command line tools

1. Check that you have **gpg (GnuPG)** package on your system.
2. If you already have a key pair, use the command `gpg --list-keys` to locate it.

```
$ gpg --list-keys
--------------------------------
pub   2048R/F2061AC8 2019-01-19
            ^^^^^^^^---------------------------------------- the key ID we need.
uid                  Picolo Developer <developer@picolo.net>
sub   2048R/62E3DF34 2019-01-19 <<-------------------------- make sure there is a 'sub' line
too.

pub   2048R/C7A7F8E4 2018-12-30
uid                  OpenPGP signature key for the architect (created on 30th Dec 2018)
<architect@matrix.com>
```

3. If you do not have a key pair yet, use the command `gpg --gen-key` to create one.
4. Enter a name and an email address for the key pair creation.

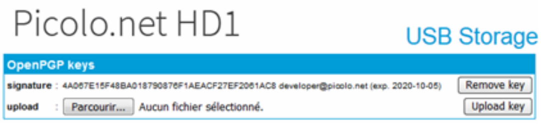
   We tested the key pair generation with 'RSA and RSA' keys and 2048-bit keys.

   When requested, enter a passphrase. This passphrase is used only on your local computer to keep the private key safe on your disk.
5. **gpg** produces a `pub/uid/sub` text block identical to that of `--list-keys output`.
6. Use `gpg -a --export THE_KEY_ID > FILENAME.asc` to isolate the desired key.

   For example, in the example, use `gpg -a --export F2061AC8 > PicoloDev.asc`

The Picolo.net HD1 server preferably uses e-mail addresses to identify key owners.
You can identify a key without an associated e-mail address only through its hexadecimal signature.

Upload your public key on a Picolo.net HD1



signature : 4A067E15F48BA018790876F1AEACF27EF2061AC8 developer@picolo.net (exp. 2020-10-05)

upload : Aucun fichier sélectionné.

1. Open the Storage page on your Picolo.net HD1.
2. Click on the Browse... button to select your exported .asc (or .pgp or .gpg) file.
3. Click on the Upload key button to add the key to the persistent configuration of the Picolo.net HD1.
4. Check that the hexadecimal string listed on the Picolo.net HD1 storage page matches the fingerprint of your key.

   To find your key fingerprint in Kleopatra, right-click the key in the main window and select Details in the contextual menu.
5. Reboot your Picolo.net HD1 to generate and encrypt a passphrase with that key.

Your Picolo.net HD1 now generates encrypted recordings:

- □ The `secure-<run-identifier>-` prefix in the file names and the ready-GPG-AES-128 status in the Media Control panel of the Storage web page confirm the encryption.
- □ All recorded files are encrypted until you remove all the **OpenPGP** keys uploaded on the device.
- □ The keys are only valid until their expiry date. When a Picolo.net HD1 in **OpenPGP+AES** mode finds only expired keys, it does not start new recordings.

2.2. Working with Encrypted Files

Every time the recording process starts (at boot time or manually), a new 'run' identifier is defined, based on the contents of the USB media plugged in.

The current AES passphrase is copied in a separate folder of the USB media and encrypted in a GPG file with the **OpenPGP** keys installed.

Retrieve the passphrase files

Retrieve the GPG files to decrypt in one of the following ways:

- Click and download the files from the web interface, on the Storage page, in the Encrypted Passphrases list.
- Open a File Explorer and go to the passphrases folder on the USB drive plugged into your computer

You can now decrypt the passphrase files using **Gpg4Win** or GPG command line tools.

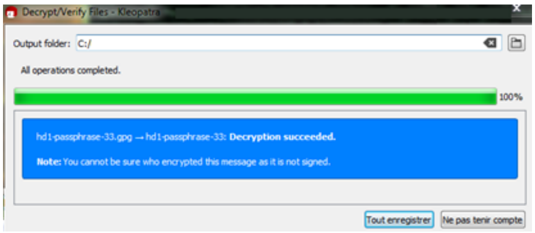
Decrypt the passphrase file using Gpg4Win

1. Select one or more GPG files.
2. Right-click on one of them and select Decrypt and verify in the menu.

   Kleopatra opens and prompts you for the passphrase you used to protect your private key.
3. Enter the passphrase.

   Kleopatra decrypts the files in the background.

4. In the Decrypt/Verify Files dialog box, select the Output folder for the decrypted files.
5. Click Save all.



All operations completed.

hd1-passphrase-33.gpg → hd1-passphrase-33: **Decryption succeeded.**

Note: You cannot be sure who encrypted this message as it is not signed.

In the specified output folder, a new file having the same name as the original GPG file is created and contains the decrypted passphrase.

When you will decrypt the corresponding MP4 file, you will copy this AES passphrase using a standard text editor.

Decrypt the passphrase file using the GPG command line tools

1. Type the command `gpg --decrypt HD1_PASSPHRASE_NN.GPG` from the folder where the passphrase files are stored.
2. Enter the private key passphrase you used to protect your private key.

The console displays the decrypted AES passphrase.

Retrieve the MP4 files

Retrieve the MP4 files to decrypt in one of the following ways:

- from the Stored Media list on the Storage page
- from the Media Preview page
- from the USB drive after plugging it in your computer

You can now decrypt the MP4 files.

Decrypt the MP4 files using Euresys decryptor sample program

1. Go to the folder containing your encrypted MP4 file.
2. Next to your MP4 files, copy the executable `decryptor.exe` you find in the sample programs on the Euresys download area for picolo.net HD1.
3. Open a command line window.
4. Open the relevant decrypted passphrase file and copy the passphrase.

5. Run the command `decryptor.exe PASSPHRASE FILENAME > OUTPUT_FILE`, where:
   - □ `PASSPHRASE` is the decrypted AES passphrase you have copied from the decrypted passphrase-%.gpg file.
   - □ `FILENAME` is the encrypted MP4 file.
   - □ `OUTPUT_FILE` is the name of the MP4 file to be generated with the decrypted video content from the encrypted MP4 file.

Decrypt the MP4 files using ecryptfs on a Linux computer

Assuming that the unencrypted file is in DIR:

1. Run the command `sudo mount -t ecryptfs DIR DIR -o`
2. Run the command `ecryptfs_unlink_sigs,ecryptfs_passthrough,ecryptfs_key_bytes=16,ecryptfs_cipher=aes,ecryptfs_enable_filename_crypto=n,no_sig_cache`
3. Enter the relevant decrypted passphrase.

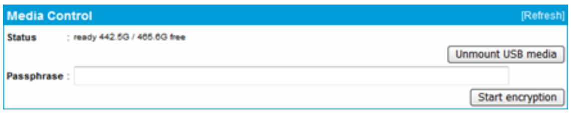
You have access to the files in their decrypted form in DIR without creating decrypted copies on the underlying media.

See https://wiki.archlinux.org/index.php/ECryptfs for alternate usages, including performing actions without root privilege.

2.3. Controlling the Encryption Manually

On devices without **OpenPGP** keys:

1. Open the Storage web page and perform the following actions in the the Media Control panel.



Media Control

Status : ready 442.5G / 465.6G free

Passphrase :

2. Enter your passphrase.

   The passphrase is not saved on the device and you must enter it again after each reboot before resuming the recording.

3. Click the Start encryption button.

   The button changes to Stop encryption

4. Stop the encryption when requested.

5. Decrypt the generated encrypted files as usual (see "Working with Encrypted Files" on page 85).

   decryptor.exe does not use the passphrase directly as the `PASSPHRASE` argument. It first convert it to `base64` format. See the README.md file delivered with the sample programs for details.

2.4. Handling the Revoked Keys

With **OpenPGP**, you can tag your keys as revoked. You cannot use these revoked keys any more for encryption and decryption operations.

Tagging a key as revoked is useful to:

- Protect e-mail exchanges (by changing the key from time to time).
- Inform publicly that an old key should no longer be used.

With Picolo.net HD1 encryption operation, we recommend that you remove the key from the ring instead.

If you upload on your Picolo.net HD1 a revoked key that:

- matches a previously uploaded key, the Picolo.net HD1 automatically removes this key from the list.
- does not match a previously uploaded key, it does not affect the list of **OpenPGP** keys.

A Picolo.net HD1 that contains only revoked keys:

- behaves as if there were no keys at all.
- disables the automatic encryption.

3. Using X.509 Certificates

3.1. About Certificates

Self-Signed certificate vs. CA-signed certificate

The Picolo.net HD1 is shipped with an on-board self-signed web certificate, that is a certificate where the public key included in the certificate is the only one validating the certificate.

Self-signed web certificates will only guarantee the integrity of the HTTPS connections. Their use may require disabling the certificate verification process of the client software.

To offer fully secured HTTPS connections that would guarantee both the integrity and the authenticity of the Picolo.net HD1 the client software is talking to, the HD1 certificate must be signed by a certificate authority (CA) trusted by the client software.

Possible signing processes

Signing Process through VMS

Your VMS may provide the features to get the Picolo.net HD1 certificate signed by a certificate authority.

1. Your VMS downloads the self-signed HD1 certificate from the Picolo.net HD1 with its SSL public key using the ONVIF Device service.
2. The VMS gets the certificate signed by one of its trusted CAs, either locally or online.
3. The VMS uploads back the signed HD1 certificate to the Picolo.net HD1 server using ONVIF services.

From then on, Picolo.net HD1 will present CA-signed certificate when establishing an HTTPS connection, which can be validated by the VMS.

Manual Signing Process

This process assumes that you already have a certificate for your own corporation and that the application(s) you intend to use (e.g. web browser) is configured to trust it as a root certificate.

An alternative would be to use a corporate certificate already signed by an established certificate authority, making it valid to your application without having to change its default set of root certificates.

The following steps will establish a certificate chain (chain of trust) between your application and your HD1:

1. In your HD1 web interface, generate a certificate signing request and download it.
2. Use a third-party tool to sign this request with your corporate certificate.

3. In your HD1 web interface, upload the corporate-signed certificate for your HD1.

The corporate-signed certificate is now stored on your HD1 and can be selected as the active certificate for HTTPS communications.

3.2. Signing Certificates with XCA

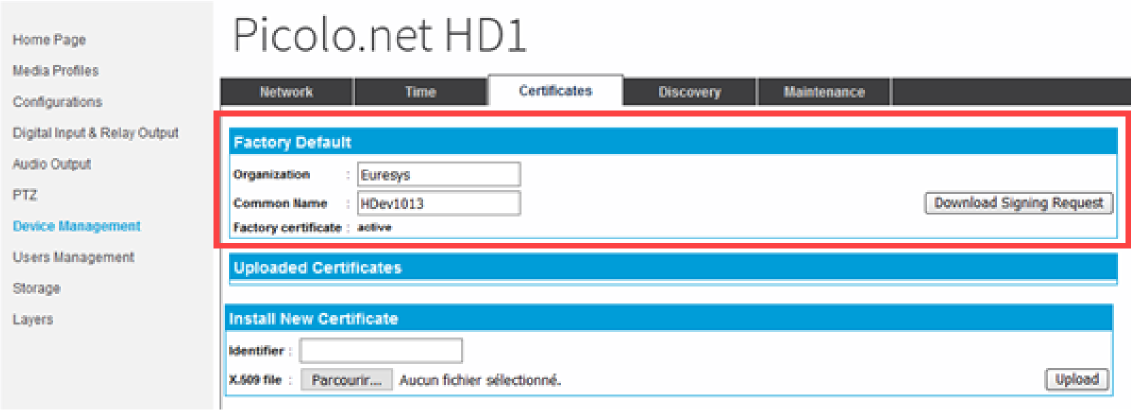
Prerequisites

Before you create your HD1 signed corporate certificate with XCA, perform the following actions:

- Download the XCA application from https://www.hohnstaedt.de/xca/index.php/download and create the initial database in which the XCA data will be stored.
- If you don't have a corporate certificate, create your own SSL private/public key pair and associate it to a self-signed certificate you define as a root CA certificate. You can use XCA to create your own key and certificate.

Generate a certificate signing request

1. Log into the Picolo.net HD1 web interface.
2. In the Device Management section, Certificates tab, fill in the fields in the Factory Default section as follows:



Organization : Euresys

Common Name : HDev1013

Factory certificate : active

Identifier :

X.509 file : Aucun fichier sélectionné.

   □ Organization: type the company name

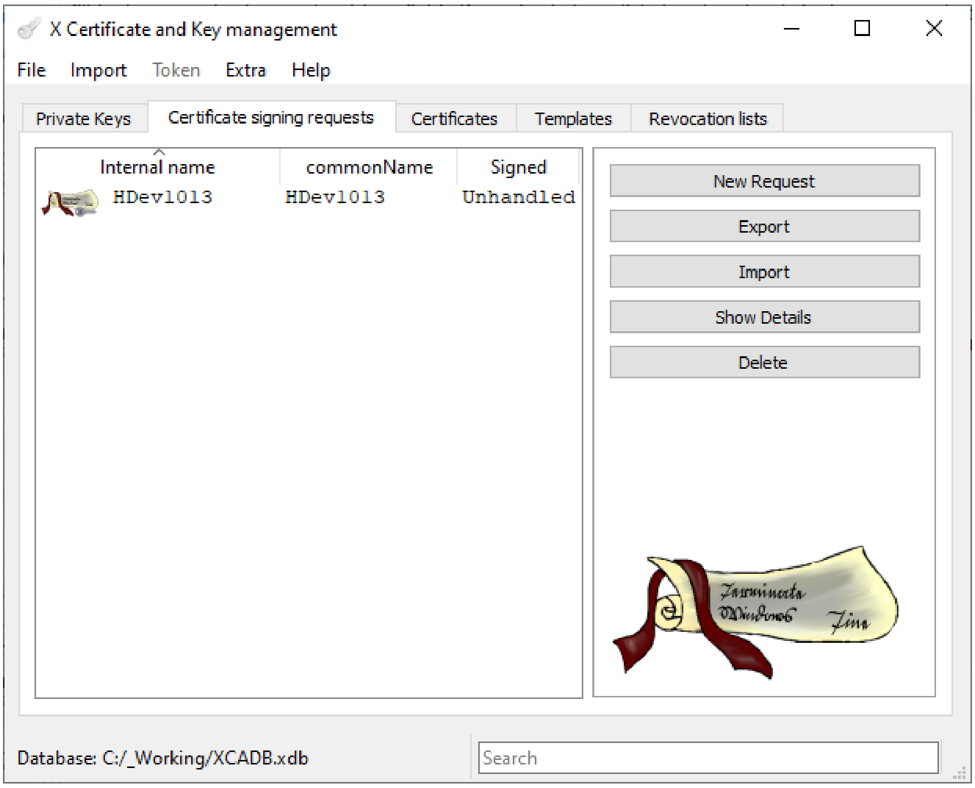
   □ Common name: type the device host name of the Picolo.net HD1 specified in the Device Management, Network tab.

3. Click Download signing request and save the generated `CSR` file locally.

   You will need this file to request the certificate signature from XCA.

Signing the X.509 certificate

1. In XCA, in the Certificate signing requests tab, click Import and select the `CSR` file you have just generated.

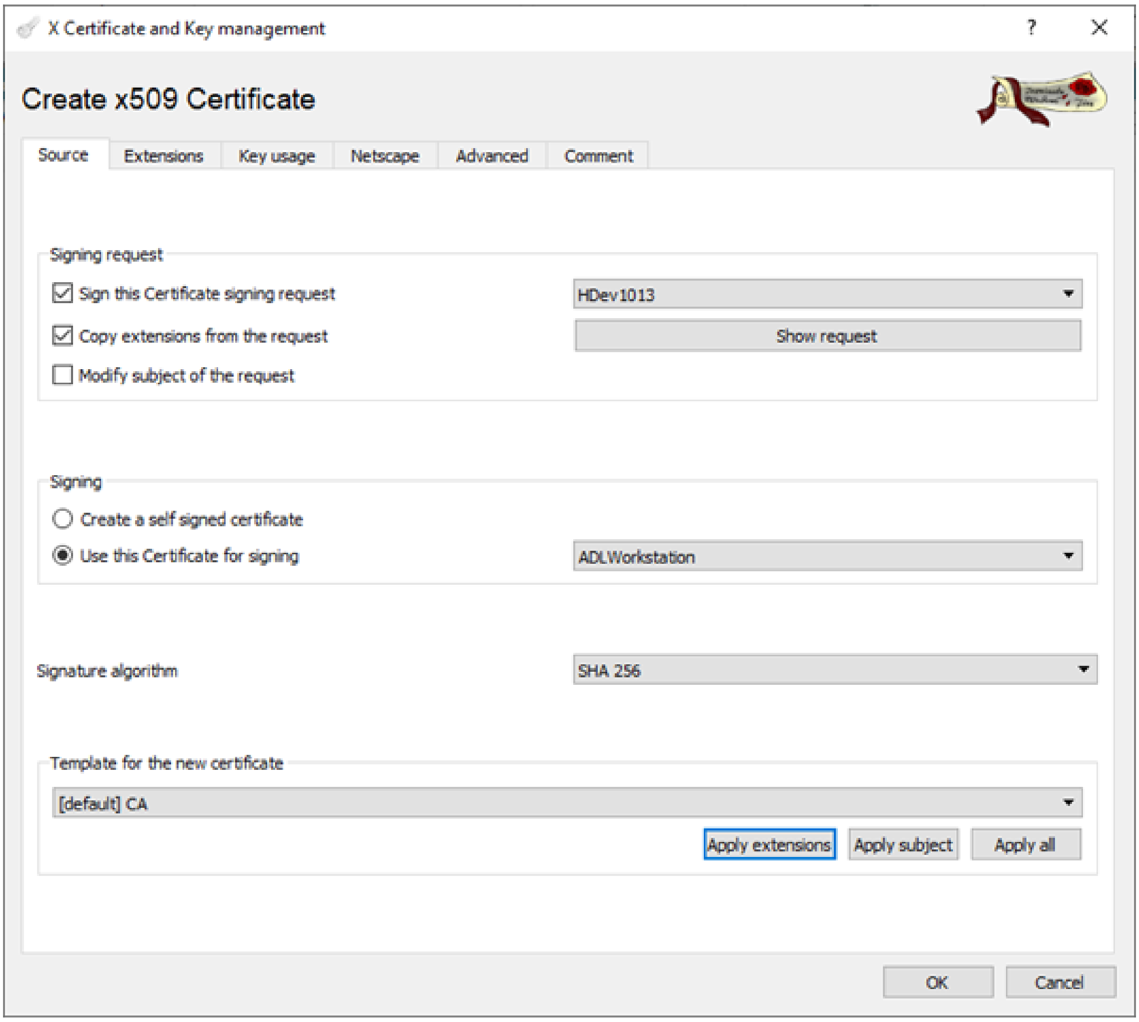


Internal name	commonName	Signed
HDev1013	HDev1013	Unhandled

Database: C:/_Working/XCADB.xdb

2. Right-click the imported certificate and select Sign from the contextual menu.

3. Fill in the fields in the Create x509 Certificate window, Source tab as follows:
   - □ In the Sign this Certificate signing request field , select the certificate signing request you have just imported.
   - □ In the Use this Certificate for signing field, select the root CA certificate (either the certificate issued by the trusted CA or the root CA certificate created in the prerequisite steps).
   - □ In the Template for the new certificate field, select [default] CA.



Create x509 Certificate

Signing request

☑ Sign this Certificate signing request: HDev1013

☑ Copy extensions from the request

☐ Modify subject of the request

Signing

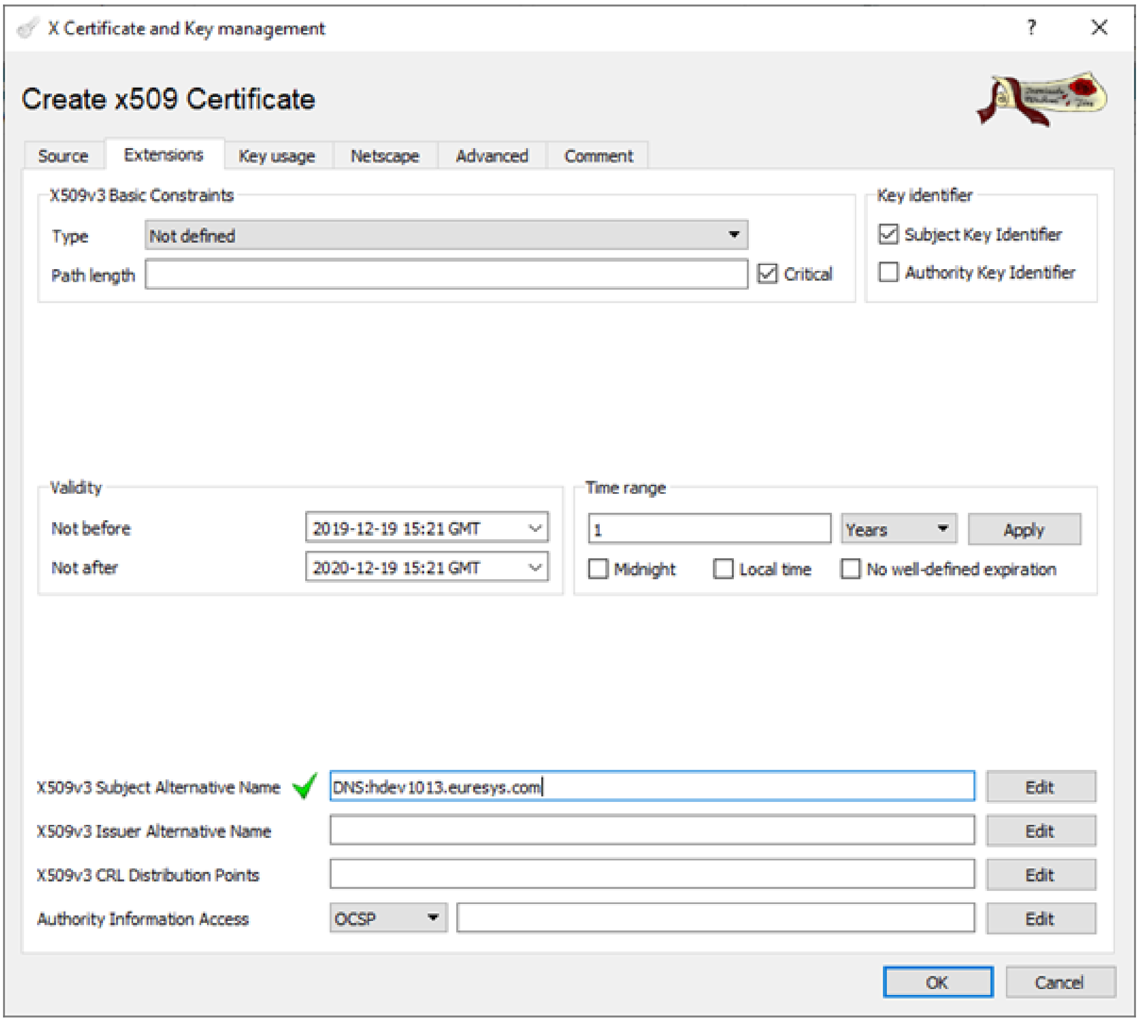
○ Create a self signed certificate

● Use this Certificate for signing: ADLWorkstation

Signature algorithm: SHA 256

Template for the new certificate: [default] CA

4. Fill the fields in the Create x509 Certificate window, Extensionstab as follows:
   - □ In the Time range area, specificy a time range for the certificate validity (at least 1 year) and click Apply.
   - □ In the X509v3 Subject Alternative Name field, specify the fully qualified domain name of your Picolo.net HD1 and prefix the string with `DNS:`



X Certificate and Key management

Create x509 Certificate

X509v3 Basic Constraints

Type: Not defined

Path length:

☑ Critical

Key identifier

☑ Subject Key Identifier

□ Authority Key Identifier

Validity

Not before: 2019-12-19 15:21 GMT

Not after: 2020-12-19 15:21 GMT

Time range: 1 Years

□ Midnight □ Local time □ No well-defined expiration

X509v3 Subject Alternative Name ✓ DNS:hdev1013.euresys.com

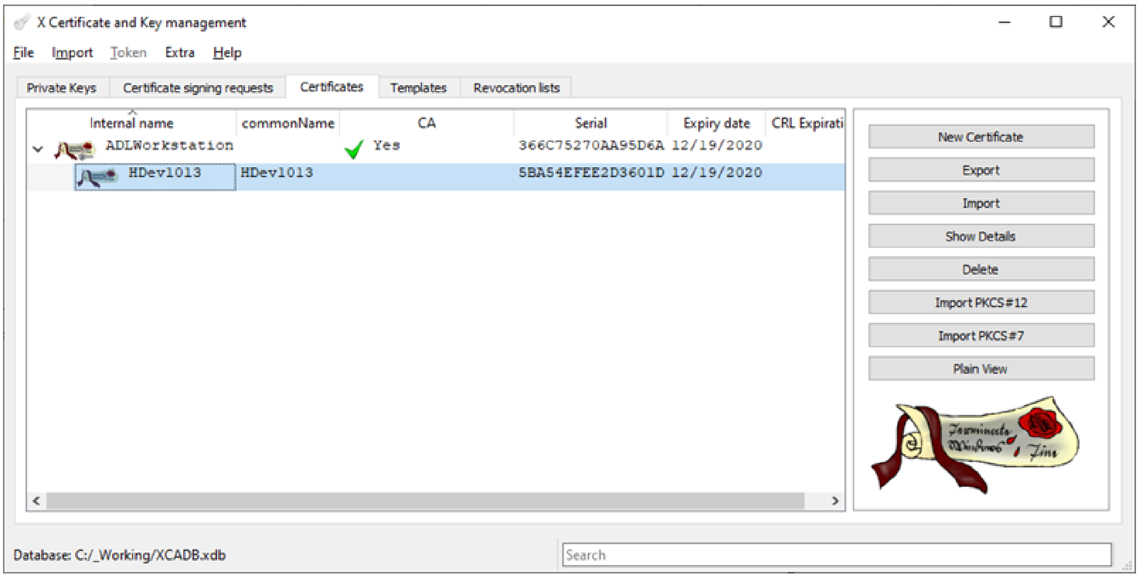
X509v3 Issuer Alternative Name

X509v3 CRL Distribution Points

Authority Information Access: OCSP

5. In the Create x509 Certificate window, Key usage tab, select the TLS Web Server Authentication key usage and click OK.

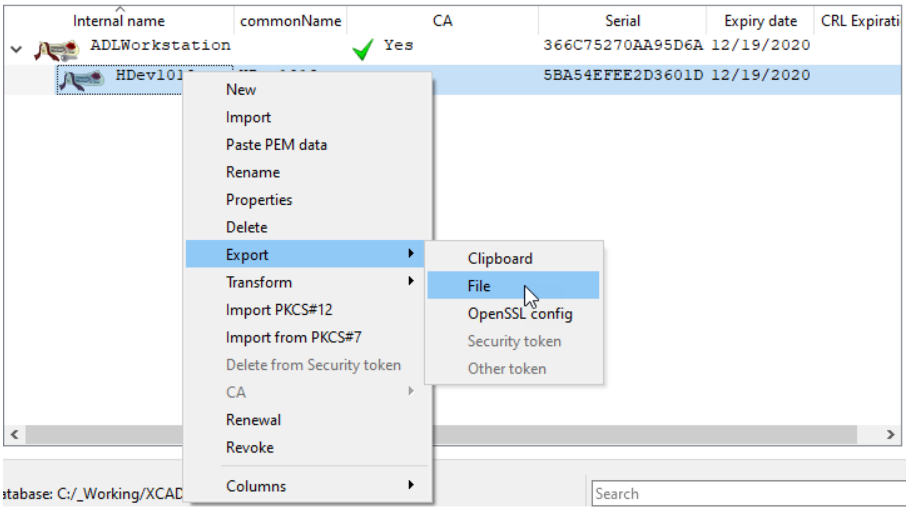
   The signed certificate should now appear under the signing (issuer) certificate in the Certificates tab of XCA application:



Internal name	commonName	CA	Serial	Expiry date	CRL Expirati
ADLWorkstation		✓ Yes	366C75270AA95D6A	12/19/2020	
HDev1013	HDev1013		5BA54EFEE2D3601D	12/19/2020	

Database: C:/_Working/XCADB.xdb

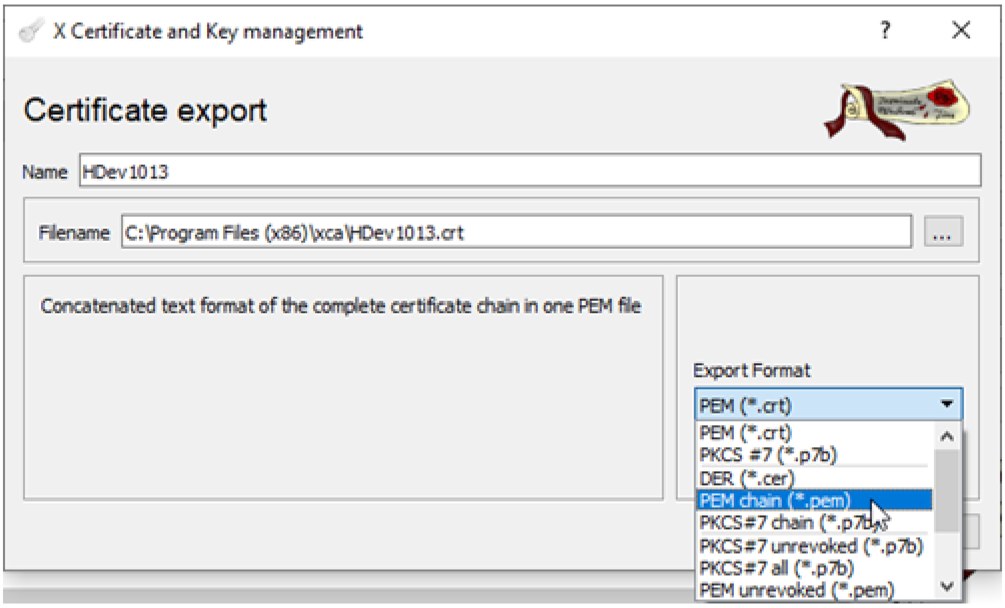
6. In the Certificates tab, right-click the HD1 certificate and select Export > File from the contextual menu:



Internal name	commonName	CA	Serial	Expiry date	CRL Expirati
ADLWorkstation		✓ Yes	366C75270AA95D6A	12/19/2020	
HDev101			5BA54EFEE2D3601D	12/19/2020	

tabase: C:/_Working/XCAD

7. In the Certificate Export dialog box, change the Export format to PEM Chain (*.pem) and click Export.



X Certificate and Key management

Certificate export

Name: HDev1013

Filename: C:\Program Files (x86)\xca\HDev1013.crt

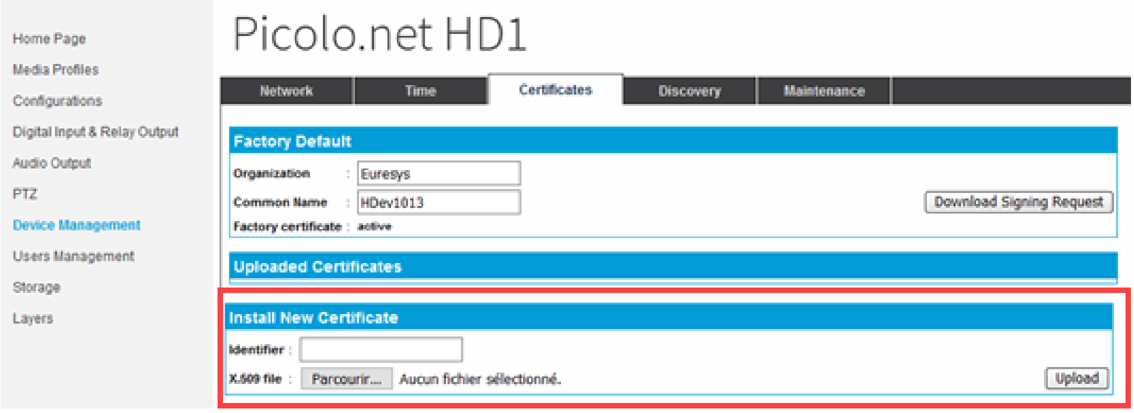
Concatenated text format of the complete certificate chain in one PEM file

Export Format: PEM (*.crt)

The PEM file is exported to C:\Program Files (x86)\xca.

Upload the X.509 certificate on the Picolo.net HD1

- In the Picolo.net HD1 web interface, in the Device Management section, Certificates tab, browse to the PEM file you have just created and click Upload to import it to your Picolo.net HD1.



Factory Default

Organization : Euresys

Common Name : HDev1013

Factory certificate : active

Uploaded Certificates

Install New Certificate

Identifier :

X.509 file : Aucun fichier sélectionné.

The new certificate is specified in the list of uploaded certificates and has the status active.

4. Connecting an External USB Drive

Using an external USB drive

The Picolo.net HD1 accepts external USB 2.0 drives formatted as unjournaling FAT32, ext2 or journaling ext3, ext4 or xfs.

1. Connect your external USB drive on one of the Picolo.net HD1 USB connectors.

Each USB connector can deliver up to 2.5 W (500 mA) to power the USB drive. If your drive requires more than 2.5 W of power, make sure you use its alternate power supply.

2. Select the USB drive operating mode:
   - **Continuous storage**: the Picolo.net HD1 captures the content until the partition is fully used.
   - **On-event storage**: the Picolo.net HD1 captures the content but keeps only the video sequences that cover a state change on the alarm GPIO connector.
   - **Rotation storage**: the Picolo.net HD1 captures the content but keeps only the N last video files.
   - **No storage**: the Picolo.net HD1 does not write on the USB drive.
3. Connect to your Picolo.net HD1 as an authenticated operator to:
   - Download the content of the active partition over HTTPS.
   - Force closing the currently active partition, as if it is full.
   - Download the device public key.

The default master directory for the files generated by the Picolo.net HD1 with the serial number `xxxxx` is `/VIDEO/NOAxxxxx` (in the root-location).

Partition selection

Partition selection sequence

When you plug a USB storage device, the Picolo.net HD1 proceeds as follows:

- If there is no partition table (that is, is the file system spans over the whole media):
  a. Picolo.net HD1 tries to mount the whole media as EXT4.
  b. If that doesn't work, it tries to mount the whole media as exFAT or FAT32.

- If there is a GUID partition table:
  a. The Picolo.net HD1 scans the partition type identifiers of the available partitions.
  b. It selects the first partition with "Linux Filesystem Data" (0FC63DAF-*) or "Windows basic data" (EBD0A0A2-*).
  c. If the selected partition is "Linux Filesystem Data", the Picolo.net HD1 only tries to mount the partition as an EXT4 file system.
- If there is a Master Boot Record:
  a. Picolo.net HD1 selects the partition type of the first partition with a FAT or a native Linux type.
  b. If the partition type is a native linux, Picolo.net HD1 only considers an EXT4 file system.

Additional precautions

> **WARNING**
> When multiple disks are available, Picolo.net HD1 arbitrarily selects one drive to store data.

- The Picolo.net HD1 can refuse to operate a partition that needs maintenance.
- If you install a supported file system on a partition with a misleading type, the Picolo.net HD1 does not use this partition.

5. Sending Custom Commands over the Serial Port

See also: The ONVIF Device IO Service | "Using the Device IO Service" on page 74 in the Reference manual

Prerequisite

- Check that the serial port is configured to match your device capabilities on "PTZ Page" on page 37.

Using the ONVIF Device IO functions

We recommend this approach in most use cases.

Use "The Device IO Functions" on page 74 to send and multiplex custom and standard (such as PELCO PTZ) commands to your device.

- Your third-party client uses the URI reported by `GetServices` response to connect to The ONVIF Device IO Service .
- Send a `SendReceiveSerialCommandRequest` to your Picolo.net HD1 to transfer a message to the serial device.
  - □ If you expect a reply from the serial device, enter the number of bytes in the `DataLength` argument of the request.
  - □ If you expect binary, use the `<Binary>` element within `<SerialData>`.
  - □ Picolo.net HD1 automatically replies with binary transport when the request uses binary.

Using `tty.php`

- The operation of tty.php is subject to modifications, so we do not recommend to use it third-party application.
- Use tty.php only for quick tests.
- Before using tty.php:
  - □ Authenticate your HTTP client from the login web page.
  - □ Check that it includes the session cookie in its requests.

Using the Euresys Remote Serial Protocol (ERSP)

The ERSP is a light TCP-to-serial bridging protocol:

- It provides a functionality similar to `SendReceiveSerialCommand`.
- It is designed for a client software that runs on a low-resource device.
- You have to authenticate only once when you launch the ERSP server.

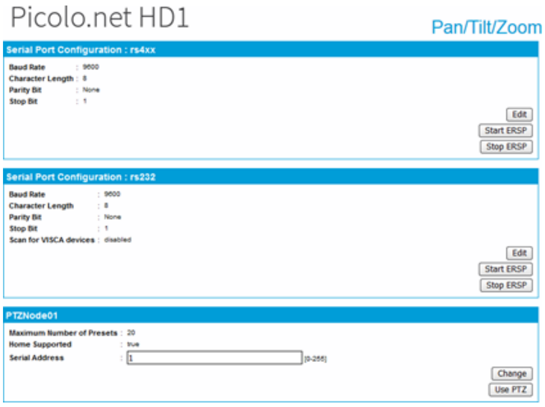
The ERSP is not available over TLS, only over plain TCP.
You should use it only when you consider that the connection between the software client and the Picolo.net HD1 is physically secure.

Fully integrated approach

1. Send a `GetTransparentSerialURI()` command to the proprietary PTZ service.
2. Extract the IP address, the port number and the authentication code from the returned URI (as illustrated in the remoteSerial.py sample program).
3. Send and receive data (as illustrated in the sample program).

Prototyping approach

1. Adapt the remoteSerial.py sample program to perform the data exchange that you want to test.
2. Use the Start ERSP button in the Serial Port Configuration panel of "PTZ Page" on page 37 to launch the server.



Picolo.net HD1

Pan/Tilt/Zoom

Serial Port Configuration : rs4xx

Baud Rate : 9600
Character Length : 8
Parity Bit : None
Stop Bit : 1

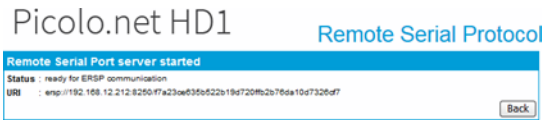
Serial Port Configuration : rs232

Baud Rate : 9600
Character Length : 8
Parity Bit : None
Stop Bit : 1
Scan for VISCA devices : disabled

PTZNode01

Maximum Number of Presets : 20
Home Supported : true
Serial Address : 1 [0-255]

3. Copy the URI starting with ersp:// from the reply web page.



Picolo.net HD1

Remote Serial Protocol

Remote Serial Port server started

Status : ready for ERSP communication
URI : ersp://192.168.12.212:8250/f7a23ce635b622b19d720ffb2b76da10d7326cf7

4. Use it to invoke the Python remoteSerial.py URI.

The ERSP server automatically terminates when the client closes the TCP session.

PART IV

APPLICATION NOTES

1. Encrypted Media Storage

Describes structures and algorithms used to offer the AES-protected storage on USB media feature of the Picolo.net HD1.

1.1. Purpose

This application note describes the cryptographic chain used by Picolo.net HD1 to protect media content stored on external USB drives against content theft or forgery.

The protection relies on a two-fold encryption mechanism:

- The **AES** layer allows the encryption and decryption of media files. The **AES** keys are derived from a passphrase generated at every boot.
- The **OpenPGP** layer allows the encryption and decryption of the **AES** keys. The **OpenPGP** encryption standard is based on pairs of private and public keys. The two matching private and public keys revert each other actions.

To access the media files on the USB drive, you have to use the private OpenPGP key matching the public OpenPGP key used to encrypt the AES keys on the Picolo.net HD1. The media files are then automatically decrypted with the appropriate AES passphrases.

1.2. eCryptfs Encryption Layer

Picolo.net HD1 takes advantage of kernel-integrated cryptography to encrypt media on-the-fly as they are written on external USB storage by means of the *eCryptfs*[1] kernel module. Files will then be encrypted with either AES-128 or AES-256, using Cipher Feedback mode (CFB), each file with its own “session key” (known as the File Encryption Key or FEK in eCryptfs codebase and documentation [1]).

[1] *www.ecryptfs.org*

One file, one key

Granting each file its own decryption key makes decryption of a new file hard even for an attacker who has access to a large stock of previously encrypted files and their decrypted counterpart. In order to keep the decryption manageable, eCryptfs does not presume that the recipient of the files will know all those keys, but instead encrypts the key with a “master key” (the File Encryption Keys Encryption Key – or FEKEK) according to the well-established PGP algorithms [3] (as described in IETF RFC2440).

To break the master key (and be able to decrypt a new file), the attacker would now need a large stock of session keys, both encrypted and decrypted.

Transparent File Management

Unlike many cryptography file-systems, which encrypt or decrypt blocks of the disk device, eCryptfs is an overlay that can be applied on any file-system technology (preferably with long file name support). This means files can still be moved, archived, organized, keeping their name and timestamps, shared to other systems and still be decrypted because each file is an autonomous container with the encrypted data and information on how to decrypt it for the intended recipient.

While the decryption process appears as “mounting” a folder in the file system on Linux platforms, it is perfectly possible for third-party applications to perform the same operations using a PGP library and some knowledge about the layout of eCryptfs files.

Simple management with Passphrase Mode

The most convenient mode of operation of eCryptfs consists in producing the master key internally from a character string known as the pass phrase. A passphrase being just a longer version of a password. PGP algorithms feature string-to-key functions that will combine hashing and cryptographic functions to produce a high-entropy, constant-sized key from that phrase, and ensure that the reverse is impossible to get. To make brute force attacks harder, some steps of that string-to-key are repeated multiple times.

1.3. eCryptfs Header

Byte address	Content	Usage
0-7	Unencrypted file size	Generic management
8-15	eCryptfs special marker	Identification
16-19	eCryptfs flags	Identification
20-23	eCryptfs extent size	Generic management
24-25	eCryptfs header extents count	Generic management
26-(xx)	RFC2440 authentication token packet set	Cryptography

Byte address	Content	Usage
(xx)-(HS-1)	Reserved	
HS-eof	Encrypted data	Payload

The table above describes the layout of the information found in the eCryptfs header section, providing cryptographic material and generic information about the file.

- The payload of the file is divided into "extents" (blocks of fixed size that can be individually encrypted or decrypted on demand).
- The "extent size" indicates how large in byte extents are in this file.
- The "header extents count" indicates how much extents containing header/padding information there are before the encrypted data.

Below is a hexadecimal dump of an eCryptfs file:

```
00000000  00 00 00 00 00 00 00 12
          /* unencrypted file size */
                                   0d 8f e7 a8 31 0e 50 5d
                                   /* eCryptfs special marker*/
00000010  03 00 00 02
          /* flags */ -- file format version == 03
                      -- properties = IS_ENCRYPTED
                      00 00 10 00
                      /* H.E.S.*/ -- Extent Size (big-endian)
                                   00 02 -- # of headers extents

RFC2440 authentication token packet set> 8c 1d 04 07 03 01

00000020  00 11 22 33 44 55 66 77  60 da 4c 8e f7 92 60 08
00000030  61 c3 9d 59 09 73 d9 83  c4
                                      ed 16 62 08 5f 43 4f
00000040  4e 53 4f 4c 45 00 00 00  00 5a 4a 2d 2e 49 56 73
00000050  f1                          /** key signature */
             00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
*
00002000 <encrypted data starts here: 2 * 0x1000>
```

Is it an eCryptfs file ?

Read the special marker in the header extent. It contains two 32-bit, big-endian words *w0* and *w1* such that *XOR(w0, w1) == 0x3c81b7f5*. This is the signature of an eCryptfs file. You can then check the flags in bytes 16 (file format version, expected to be 3) and 19 (properties, bit 1 set indicates an encrypted file).

Where does the encrypted payload start ?

The payload starts after <*header extents count*> blocks of <*extent size*> bytes. Here, this is 2 times 4096 (0x1000) bytes. We also know that the file will be only 18 bytes (<*unencrypted file size*>) once decrypted. Because eCryptfs uses block cryptography, this can be smaller than the size of the encrypted data section.

How to get the session key ?

The cryptography material is contained in the variable-sized *authentication token packet set* chunk, starting at offset 26 in the header extent. Each packet in this set starts with a type byte followed by one or more size byte(s) and finally some payload bytes, as defined in section 4.3 of RFC2440.

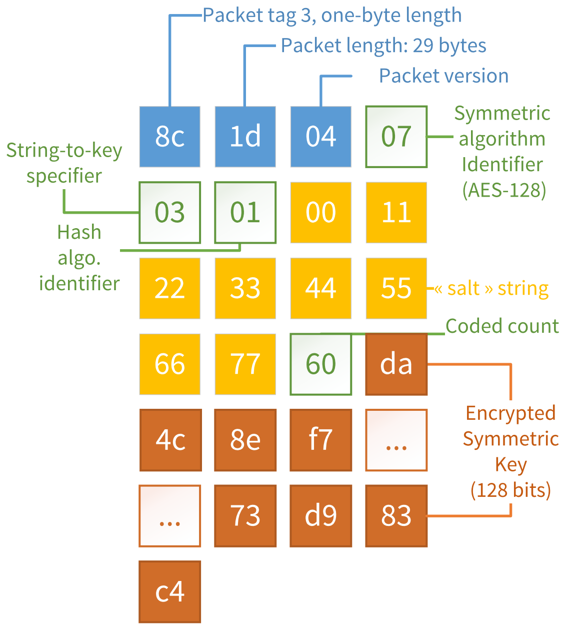
The current firmware for Picolo.net HD1 only supports passphrases to authenticate users of the encrypted storage, meaning that the only two packets expected in the set are:

- Symmetric key encrypted (packet tag 3), as defined in section 5.3 of the RFC
- eCryptfs key signature (packet type 0x2d) following the generic “literal data” structure (packet tag 11) as described in section 5.9 of the RFC.

The key signature uniquely identifies the master key used to encrypt the session key contained in the tag-3 packet. eCryptfs uses it to look up for the corresponding key in its internal keyring.

According to §5.3 of the RFC:

- *If the encrypted session key is present, the result of applying the S2K [string to key] algorithm to the passphrase is used to decrypt just that encrypted session key field, using CFB mode with an IV [initialization vector] of all zeros.*
- *The decryption result consists of a one-octet algorithm identifier that specifies the symmetric-key encryption algorithm used to encrypt the following Symmetrically Encrypted Data Packet, followed by the session key octets themselves.*



```
8c 1d 04 07
03 01 00 11
22 33 44 55
66 77 60 da
4c 8e f7 ...
... 73 d9 83
c4
```

On the sample encrypted file, we can tell from the “string to key” specifier that we will have to use “iterated and salted” algorithm (specifier #3, described at section 3.6.1.3 of the RFC) using the SHA-512 hash algorithm (identifier #1), 65536 times.
This is as currently used in the Linux kernel implementation and it departs from the RFC2440 specifications.

The first 128-bit of the resulting hash gives us the “master key” required to decrypt the session key as indicated above.

1.4. Web Services

The Picolo.net HD1 device can have its media store either *locked* or *unlocked*:

- When the store is *locked*, the AES layer is disabled: files previously written are not decrypted and newly recorded clips are stored as plain MP4 files.
- When the store is *unlocked*, the AES layer is enabled: AES decryption is applied to previous files and AES encryption is applied to incoming clips recorded by the device.

The `LockAESStorage` and `UnlockAESStorage` methods of the `HD1RecordingProprietaryService` on the Picolo.net HD1 allow automation of the switching between the two states of the media store.

Those services are complemented with a `GetAESStorageStatus` call that can be used to read the current state of the media storage directory and its name on the USB media.

Relevant XSD elements

```
<xs:element name="UnlockAESStorageRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="PassPhrase" type="xs:string"/>
            <xs:element name="Directory" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

The `UnlockAESStorageRequest` message provides the passphrase as plain string and the directory on the USB media that should be mounted (normally <device-serial-number>.<encoder-identifier>).

NOTE
Using the `UnlockAESStorage` method transmits the passphrase in clear text on the network unless the caller has established an https session to deliver its request.

NOTE
Given the amount of computations needed to convert the passphrase into the appropriate master key, receiving the `UnlockAESStorageResponse` doesn’t guarantee that the directory is effectively protected. To confirm that the directory is protected, the caller should invoke `GetAESStorageStatus` and test the ‘encryption’ field.

```
<xs:simpleType name="EncryptionEnum">
    <xs:restriction base="xs:string">
       <xs:enumeration value="None"/>
       <xs:enumeration value="AES128"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="GetAESStorageStatusResponse">
  <xs:complexType>
        <xs:sequence>
            <xs:element name="Directory" type="xs:string"/>
            <xs:element name="encryption" type="tns:EncryptionEnum"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

1.5. References

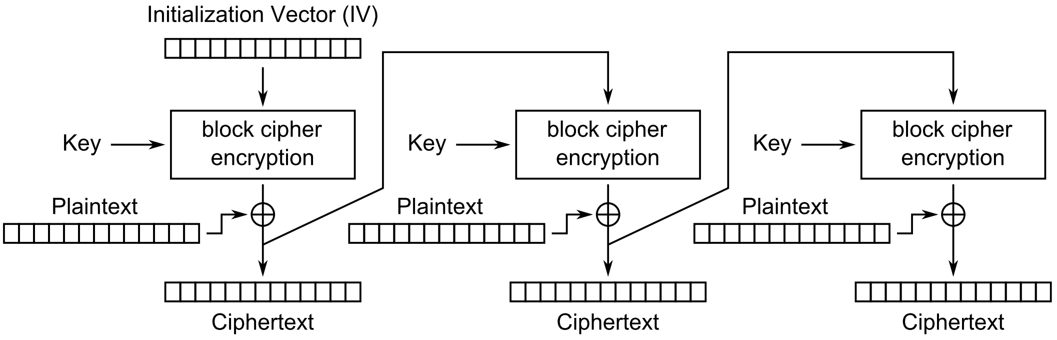
1. https://www.kernel.org/doc/Documentation/security/keys-ecryptfs.txt
2. https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
3. https://www.ietf.org/rfc/rfc2440.txt

1.6. Appendix

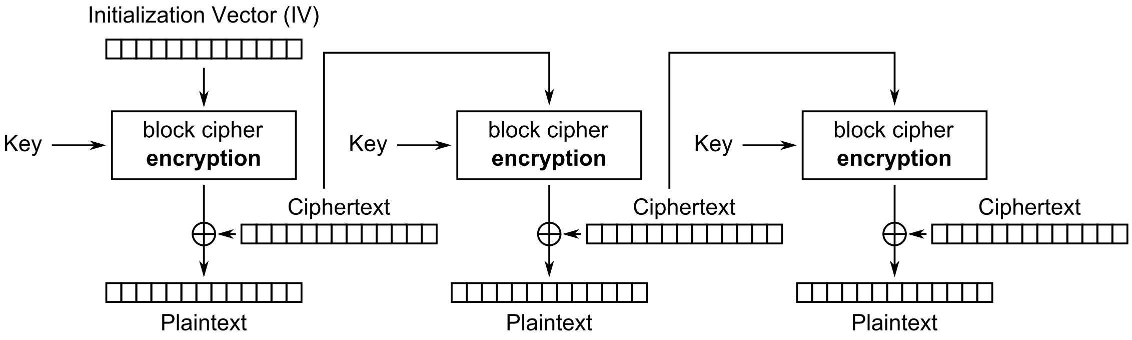
Cipher Feedback Mode for AES

Source: Wikipedia [2]

The cipher feedback mode - CFB - has the desirable properties that large, continuous blocks of identical data cannot be easily recognized but still allows to decrypt at random locations in the stream since only the previous ciphered text – not plain text – is required in addition to the key to decrypt a given block.



Cipher Feedback (CFB) mode encryption



Initialization Vector (IV)
Key
block cipher **encryption**
Ciphertext
Plaintext
Key
block cipher **encryption**
Ciphertext
Plaintext
Key
block cipher **encryption**
Ciphertext
Plaintext

Cipher Feedback (CFB) mode decryption

2. Coding Guidelines for VMS Application

2.1. Web Services

Web services inherited from Picolo.net HD4

feature	standard body	standard	release	comments
device discovery	W3C	discovery	1.0	
list optional capabilities	onvif	core	1.0	
firmware upgrade	--	--	1.0	only through web pages
network configuration	onvif	device	1.0	static IPv4 / DHCP
configure discovery	onvif	device	1.0	
get system logs	onvif	device	1.0	
system information	onvif	device	1.0	
security configuration	onvif	device	TBD	
get temperature	picolo.net	device	TBD	
configure media profiles	onvif	media	1.0	
live media streaming	onvif	streaming	1.0	
RTP UDP/UDP multicast	ietf	RFC3550	1.0	
RTP/RTSP/HTTP	QuickTime	RTSP_Over_HTTP	1.0	
RTP/RTSP/HTTPS	-	-	1.0	reference client implementation: gstreamer-1.0
	picolo.net		1.0	definition in this document

feature	standard body	standard	release	comments
Auto-setup Profile	picolo.net	media	1.0	
Basic notification interface	W3C	wsn-b2	TBD	
Real-time Pull-point Notification Interface	onvif	events	TBD	
Notification streaming interface	onvif	events	TBD	
Configure PTZ nodes	onvif	ptz	1.0	
Continuous Pan/tilt/zoom movements	onvif	ptz	1.0	
Stop PTZ movement	onvif	ptz	1.0	
PTZ Presets management	onvif	ptz	1.0	
Configure COM port	picolo.net	ptz	1.0	
Custom messages on RS-xxx connectors	onvif	deviceIO	1.2	
Control relay output	onvif	deviceIO	TBD	
Configure alarm input	onvif	deviceIO	TBD	

Services new to Picolo.net HD1

feature	standard body	service	release	comment
start/stop recording	onvif	recording	1.0	through the SetRecordingJobMode() call. Recordings and RecordingJobs are statically defined.
define recorded clip length	picolo.net	recording	1.0	through SetExtendedRecordingConfiguration() call, using the token matching the Recording object you want to configure

2.2. Encrypted Live Stream Support

An additional web service `GetPicoloHttpsUri()` allows a client to know which URI to use to retrieve live captured media over TLS-protected connections.

The data stream carried over these HTTPS connections can be either MP4 file (containing H.264 video + AAC audio) or RTSP-tunnelled-over-HTTP, and is decided by the `format` argument passed to the service. Once the URI is retrieved by the client, the client is then responsible of setting up the proper protocol stack to communicate with the server located by the URI.

Retrieving URI example (php)

```
// let $Addresses["Media"] correspond to Media->XAddr
// in the reply to onvif's GetCapabilities() core service.
$webServiceClients["MediaProprietary"] = new
OnvifWebServiceClient("wsdl/hd4MediaProprietaryService.wsdl",
$Addresses["Media"]);
$svc = $webServiceClients["MediaProprietary"]->GetInstance();
$tunparams = array('format' => 'application/x-rtsp-tunnelled',
'ProfileToken' => $pagetoken);
$uri = $svc->GetPicoloHttpsUri($tunparams);
```

The video/mp4 format can be passed as-is to an HTML5 compliant web browser for playback. The `x-rtsp-tunnelled` format can be requested by gstreamer-1.0 and converted into raw H.264 NALUs byte stream using the following components pipeline (tested on Ubuntu LTS 14.04):

gstreamer invocation

```
gst-launch-1.0 rtspsrc 'location=$uri' tls-validation-flags=generic-error \
 !rtph264depay ! "video/x-h264,stream-format=byte-stream" \
 ! filesink location=/tmp/example.h264
```

where

- `$uri` is the result of above php example and will start with `rtspsh://` to indicate the use of RTSP-over-HTTP and secure TLS connections.
- `tls-validation-flags=generic-error` indicate to skip X.509 certificate chains
- `rtph264depay` extracts the H.264 payload out of RTP packets
- `video/x-h264,stream-format=byte-stream` forces a `\x00\x00\x00\x01` marker between H.264 NAL units
- `filesink` places the bitstream as is in a file.

Relevant XSD elements

```
<xs:simpleType name="PicoloHttpsStreamMime">
     <xs:restriction base="xs:string">
         <xs:enumeration value="video/mp4"/>
         <xs:enumeration value="application/x-rtsp-tunnelled"/>
     </xs:restriction>
 </xs:simpleType>

 <xs:element name="GetPicoloHttpsUriRequest">
     <xs:complexType>
          <xs:sequence>
               <xs:element name="format" type="tns:PicoloHttpsStreamMime"/>
               <xs:element name="ProfileToken" type="tt:ReferenceToken"/>
          </xs:sequence>
     </xs:complexType>
 </xs:element>
 <xs:element name="GetPicoloHttpsUriResponse">
```

```
     <xs:complexType>
          <xs:sequence>
              <xs:element name="MediaUri" type="tt:MediaUri"/>
          </xs:sequence>
     </xs:complexType>
 </xs:element>
```

WSDL additional definitions

```
<wsdl:message name="GetPicoloHttpsUriRequest">
     <wsdl:part name="body" element="xsd:GetPicoloHttpsUriRequest"/>
 </wsdl:message>

 <wsdl:message name="GetPicoloHttpsUriResponse">
     <wsdl:part name="body" element="xsd:GetPicoloHttpsUriResponse"/>
 </wsdl:message>
 <wsdl:portType name="PicoloMediaProprietaryPortType">
     <!-- other operations definitions -->
     <wsdl:operation name="GetPicoloHttpsUri">
         <wsdl:input message="tns:GetPicoloHttpsUriRequest"/>
         <wsdl:output message="tns:GetPicoloHttpsUriResponse"/>
     </wsdl:operation>
 </wsdl:portType>
```

2.3. Reference Documents

- https://www.onvif.org/specs/DocMap-2.3.html
    a. https://www.onvif.org/specs/core/ONVIF-Core-Specification-v230.pdf (device service, event service)
    b. https://www.onvif.org/specs/srv/io/ONVIF-DeviceIo-Service-Spec-v221.pdf
    c. https://www.onvif.org/specs/stream/ONVIF-Streaming-Spec-v220.pdf
    d. https://www.onvif.org/specs/srv/rec/ONVIF-RecordingControl-Service-Spec-v211.pdf