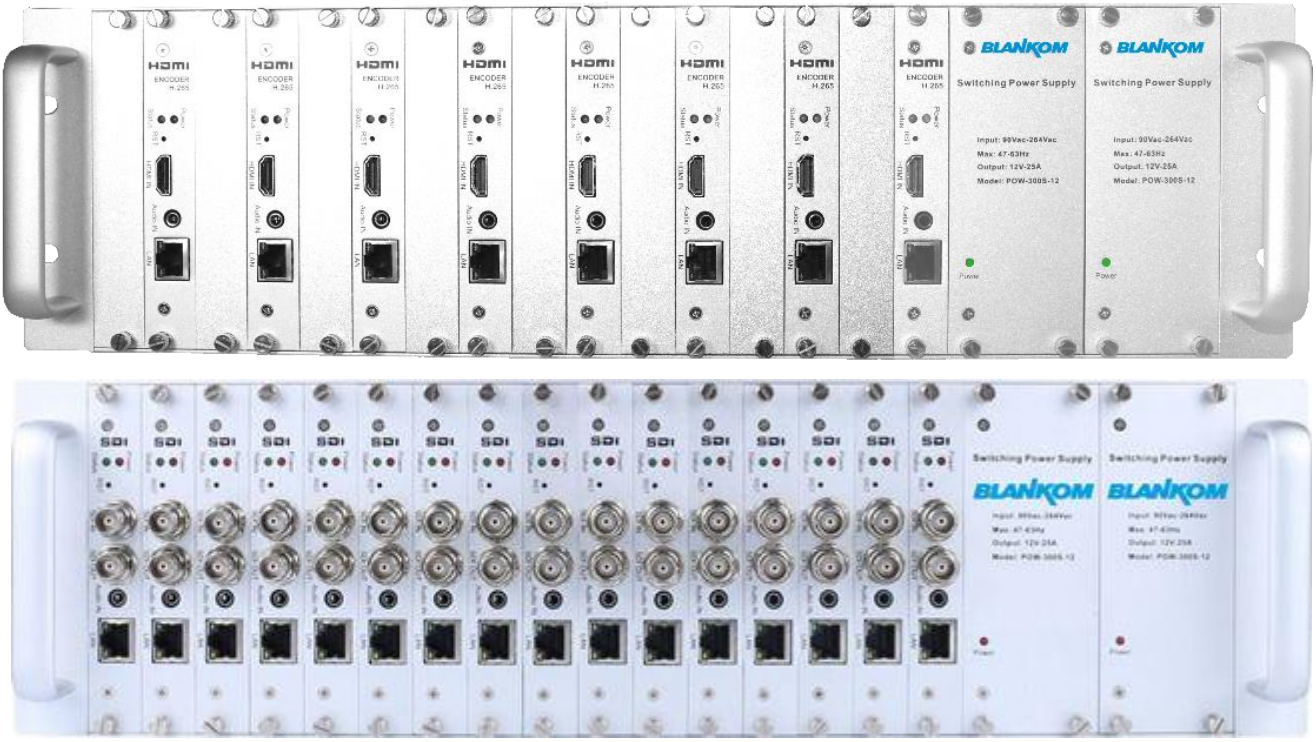
A general description for our SoC / DSP -Encoder Series

Basics and Tipps and Tricks

BLANKOM an IRENIS® Brand ...setting signals



h.264 / h.265 (HEVC)
SoC IPTV Encoder & Streamer series

Inhalt TOC

Important Notes!

This manual is for use by qualified personnel only. Handling this device or system requires special electronic technical knowledge. To reduce the risk of electrical shock or damage to the equipment, do not perform any servicing other than the installation and operating instructions contained in this manual unless you are qualified to do so. This device operates in the given voltage and frequency range without requiring manual adjustment.
Do not open the top case w/o unplugged power source because serious injury or death may be the result!
Inside are components under risk from **electrostatic discharge**. To avoid equipment damages do not touch these components or, observe the respective handling rules!
For continued protection against fire, the fuses may only be replaced by identical fuses with the same electrical specifications which are designed for the corresponding fuse positions.

IRENIS GmbH reserves the right to revise this publication and make changes in its content from time to time, whereby it shall not be obligatory for IRENIS GmbH to provide notification of such revision or change.
IRENIS GmbH provides this manual without warranty of any kind, neither implied nor expressed, this includes also any warranty's regarding the merchantability and fitness for a particular purpose. IRENIS GmbH may improve this manual or make changes in the products described herein at any point of time.

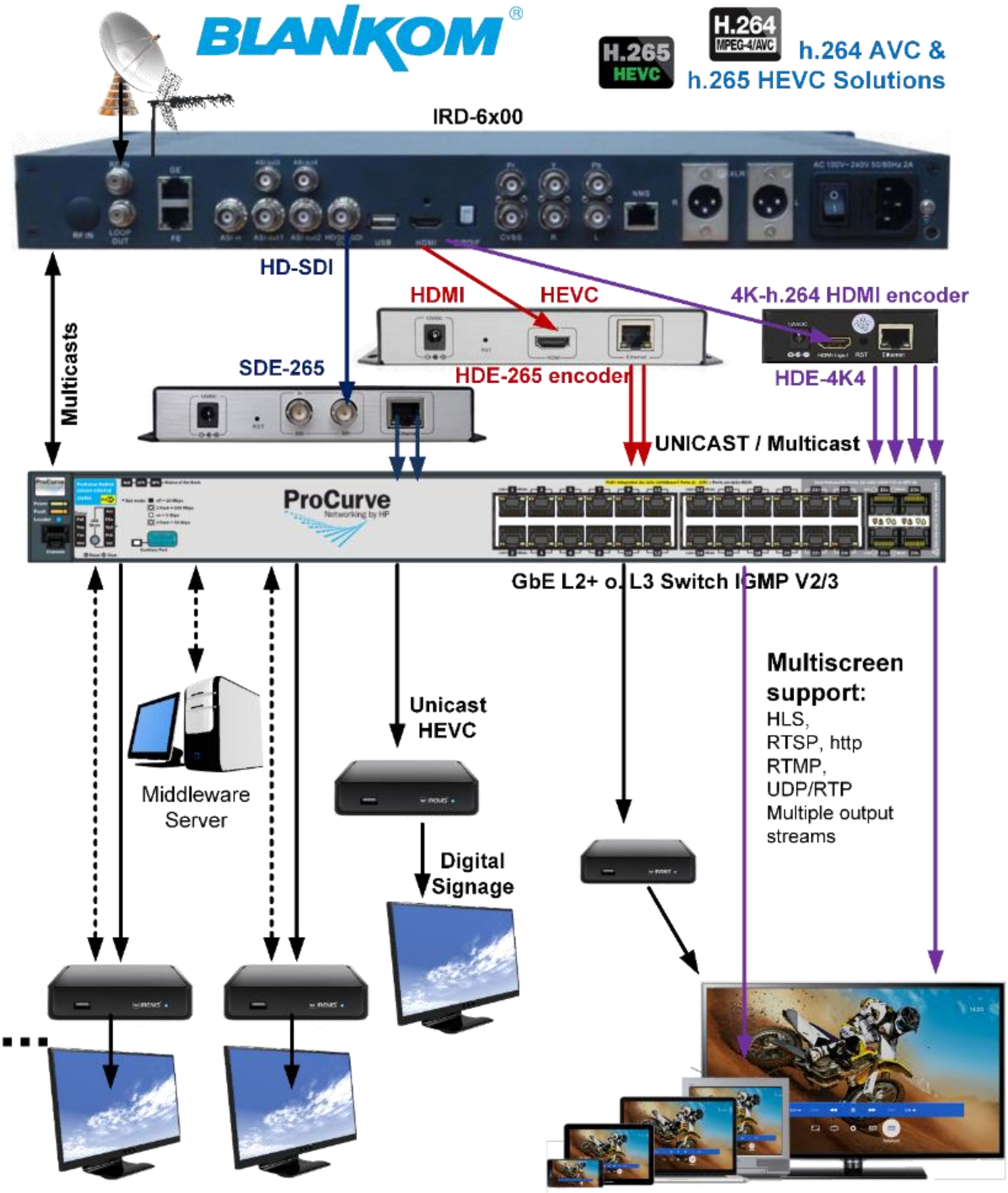
Product Overview

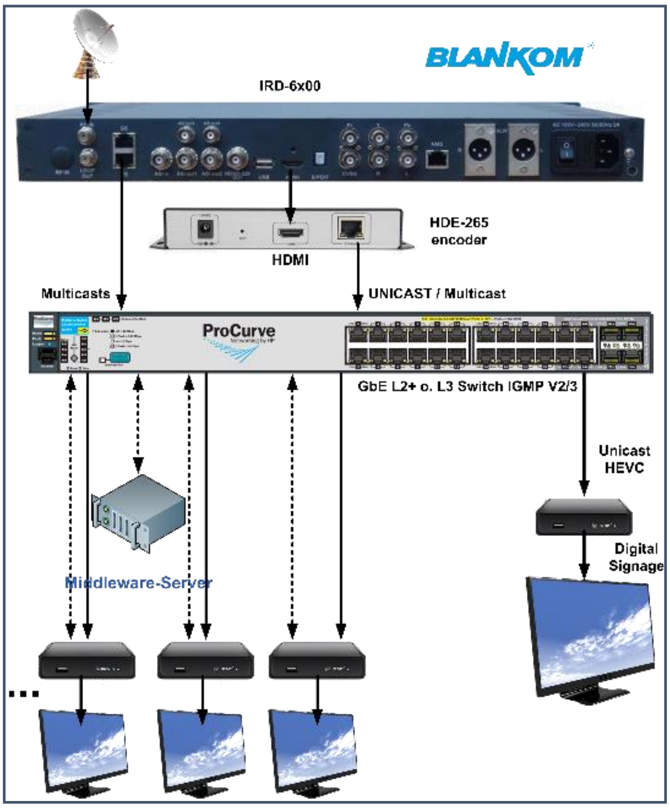
The h.264/h.265 compatible Encoder is a hardware device used for high-definition video signal (up to 1080p60 HD resolution) encoding and network transmission, using the latest and high-efficient HD digital video compression technology h.264/H.265, with the characteristics of reliable, high-definition, low bitrate and low latency. Connect the HDMI/SDI/VGA high-definition video signal to start the encoding process, after the compression processing by the DSP chip, the output of the standard TS network stream can be started.
The launch of these devices fills the gap in the industry, which is a direct replacement for the traditional capture card for software coding method, using hard-coded chipsets, the system is more stable, and the picture quality is more perfect. They can be used in a wide variety of demands for high-definition video and high-resolution / high frame rate re-assembling for IP based network transmission. Its powerful scalability makes it easier to respond to the needs of different industries and can be used as live video encoder. Industrial controlled, precision design, small size, easy installation, the power is less than 5W, which is energy-saving and more stable.

This general manual applies to the following models:

All SoC DSP based boxed Encoders SDE- HDE - and SHDE.
Where S stands for SDI, H = HDMI, SH for both in a box, D =Digital, E= Encoder, 2nd D = Decoder Version like HDD-275.

Application Example:





IMPORTAND NOTE:
Please connect your PC/Laptop and the Encoder(s) always to the Ethernet with a GbE auto-negotiation Switch (10/100/1000BaseT) in between.

Otherwise, you might damage either your laptop or the encoder RJ45 ports(s) or at least get connection problems.

Assure that your switch doesn't do Multicast-blocking on the ports you connect it – if you use UDP/RTP multicasts.

Encoder

HDE-275:

Almost all types and models have a separate data sheet and Quick start-manuals

NOTE:
A new and here eventually not listed devices like
HDE-4K5C = Wall mount version of HDE-275
4 in 1: HDE-275Q Quad-encoder:
(2x UHDp30 2x HDp60)

Counterparts:

The UHD- DECODER HDD-275 (Feature: HD-SDI output) and HDD-275H (HDMI output only)

Default Values

The factory default administrator account: *admin*
The factory-default user password: *admin*
The factory default IP address: *192.168.1.168*
Please change these account settings according to your local policy and network. -> Do not forget to safe and backup the configuration.

Basics - Unicasts:

RTSP over HTTP
The key of RTSP over HTTP is to allow RTSP packets to communicate via HTTP port.
We know that the standard port of RTSP is 554, but due to various security policy configurations such as firewalls, there may be restrictions when the client accesses port 554, which prevents the normal transmission of RTSP packets. But the HTTP port (port 80) is generally open, so there is the idea of letting RTSP packets pass through port 80, namely RTSP over HTTP.
The details of RTSP over HTTP are as follows:
First, the client opens two socket connect to the rtsp server HTTP ports. We call these two sockets "data socket" and "command socket".
Step 1. The client sends an HTTP GET command through the "data socket" to request an RTSP connection.
Step 2. The server responds to the HTTP GET command through the "data socket" and responds with success/failure.
Step 3. The client creates a "command socket" and sends an HTTP POST command through the "command socket" to establish an RTSP session.

At this point, the auxiliary function of HTTP is completed, and the server does not return the client's HTTP POST command. Next is the standard process of RTSP on the HTTP port, but it needs to be completed through two sockets. The "command socket" is only responsible for sending, and the "data socket" is only responsible for receiving.

Step 4. The client sends RTSP commands (BASE64 encoding) through the "command socket".
Step 5. The server responds to the RTSP command (in plain text) through the "data socket".
Step 6. Repeat Step4-Step5 until the client sends the RTSP PLAY command and the server responds to the RTSP PLAY command.
Step 7. The server transmits audio and video data to the client through the "data socket" After the data exchange is complete...
Step 8. The client sends the RTSP TEARDOWN command (BASE64 encoding and) through the "command socket"
Step 9. The server responds to the RTSP TEARDOWN command (in plain text) through the "data socket".
Step 10. Close the two sockets.

Secure Reliable Transport

SRT (Secure Reliable Transport) is an open-source Internet transmission protocol based on UDT protocol. Haivision and Wowza have cooperated to establish SRT alliance to manage and support open-source applications of SRT protocol. This organization is committed to promoting the interoperability of video streaming solutions and realizing low latency network video transmission.

SRT is improved from UDT (UDP based Data Transfer) protocol. SRT protocol retains most of the core concepts and mechanisms of UDT protocol, and introduces some improved and enhanced functions, including flow control for real-time audio and video, enhanced congestion control, modification of control data, and improvement of encryption mechanism.

<u>SRT protocol features:</u>
Three features: safety, reliability and low latency.
In terms of security, SRT supports AES encryption to ensure end-to-end video transmission security.
In terms of reliability, SRT ensures the stability of transmission through forward correction technology (FEC).
In terms of low latency, SRT is based on UDP protocol, which solves the problem of high transmission latency of TCP

protocol.
SRT solves the complex transmission timing problem, and can support real-time transmission of high-throughput files and high-definition videos.

Advantages of the SRT protocol:
Reliability: It is suitable for any network environment and can efficiently handle network packet loss, jitter, bandwidth fluctuation and other disturbances;
Low latency: Due to the UDP transmission mode and ARQ packet loss recovery mechanism, the transmission delay level based on the public network can generally be controlled within 1s;
High quality: SRT's transmission and error correction mechanism can maximize the use of available bandwidth and eliminate network errors and interference, so it can transmit higher bit rate video streams in the same network environment, and cooperate with H Efficient encoding formats such as 264 and HEVC can ensure high video quality under poor network conditions;
High bandwidth utilization: The multi rate adaptive distribution technology, which is different from ABR, requires additional bandwidth for redundant rate. SRT monitors the network link status in real time and can adjust the rate in real time (NAE, network adaptive coding). In addition, ARQ's packet loss recovery mechanism also greatly saves bandwidth and reduces network congestion compared with TCP's packet loss recovery mechanism;
Security: SRT uses AES-128 or 256 encryption to protect content security;
Free and open source: SRT is completely free and open source.
Shortcomings of SRT:
SRT is based on bidirectional UDP point-to-point connection, which is suitable for high-quality, low latency and reliable transmission of point-to-point, but not suitable for content distribution to mass users.

Product features

High-performance hardware encoding

- h.265 compatible video coding efficiency (depending on Model, downward compatible to h.264)*
- h.264 BP/MP/HP
- AAC / G.711 Advanced Audio Coding format quality (* MP1L2, AAC++, MP3, AC3)
- CBR / VBR encoding rate: 16Kbps ... 12Mbps
- 100BaseT or 1000Mbit/s network interface using full duplex mode (dep. on Model) *
- A mainstream and second stream can be sent to different IP-connections (HDE-4K4/5/C up to 4 streams) Supports up to 720P, 1080P @ 60HZ HD video input
- Support image parameter settings
- HTTP, HLS, UDP, RTSP, RTMPs, RTP, ONVIF protocol
- The mainstream and secondary stream(s) can be used with different network protocol for their transmissions
- WEB interface English
- Remote management in WAN/LAN (WEB)
- Support customized settings for the resolution
- Support one step to restore the factory configuration

** Features may vary between models and Software Versions*

Applications

- IPTV
- Digital Signage
- Video Conference
- Hotel IPTV System
- Live Broadcast Feeds
- Campus IPTV System
- IP Recording System
- Medical video broadcast and recording system
- Live video education system
- IP Video Recording (DVR/NVR)
- 4G mobile broadcast HD front capturing

WEB server –Access settings

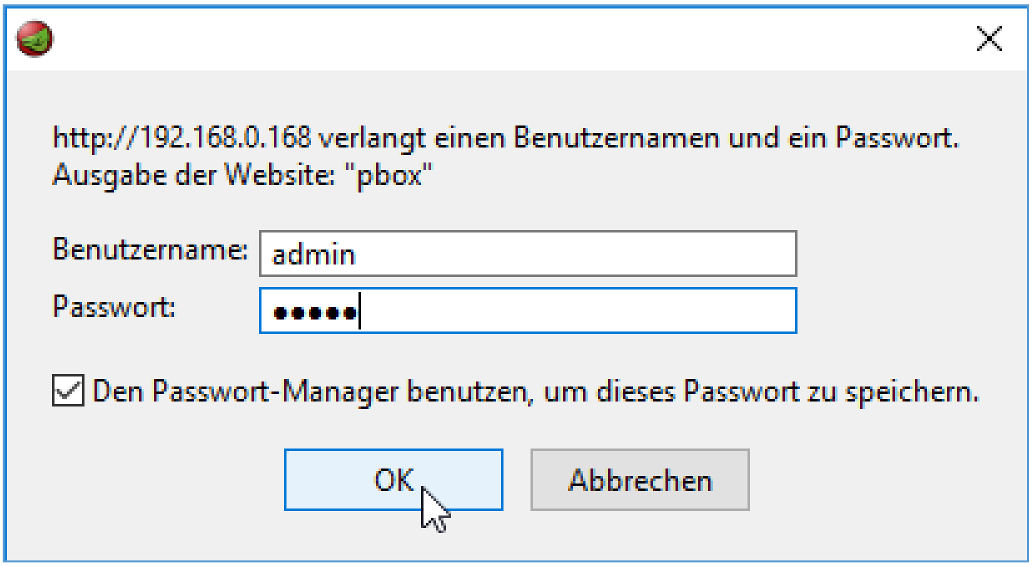
Step 1: Reset & initialization
Connect the power supply to turn on the encoder and use a pin to press RST on the encoder for min. 10 seconds (better 15), it will be restarted and initialized. The default Route IP of the WEB-IF is 192.168.1.168 (*This default Address may vary depending on model)* after initialization and can be recognized from the sticker at the bottom.

HINT if you lose your IP address, there is a 'RESET'-Button hole at the front. Press until 1 LED goes off (>10seconds).

Step 2: Change the administrator's computer IP
Set the administrator's computer IP as: 192.168.0.* or 192.168.1.*to avoid IP conflicting with the units own IP address IP 192.168.0.*: (use an IP setting "*" in the number range between 2-254 except .1.168) Remark: .0 is often the network router, .1 often the Gateway of the used router.

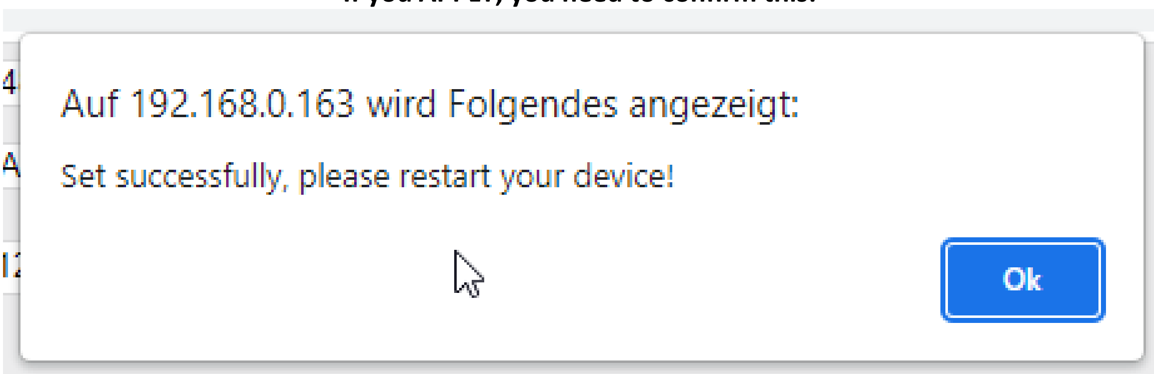
Step 3: Login the menu with the web browser:
Enter 192.168.1.168 in your Browser window.

Default user Name: admin
Default password: admin



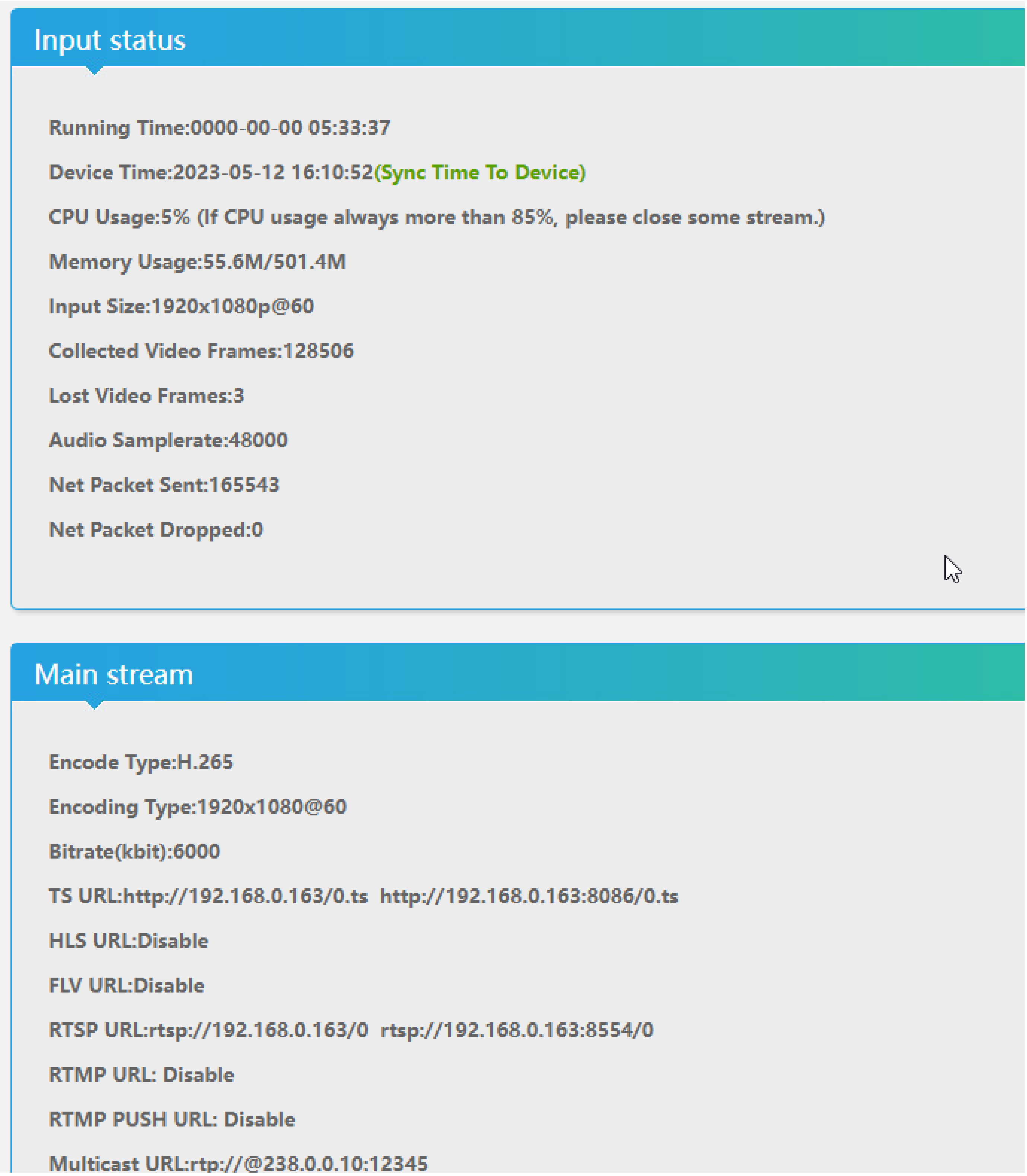
A remark upfront:

If you APPLY, you need to confirm this:

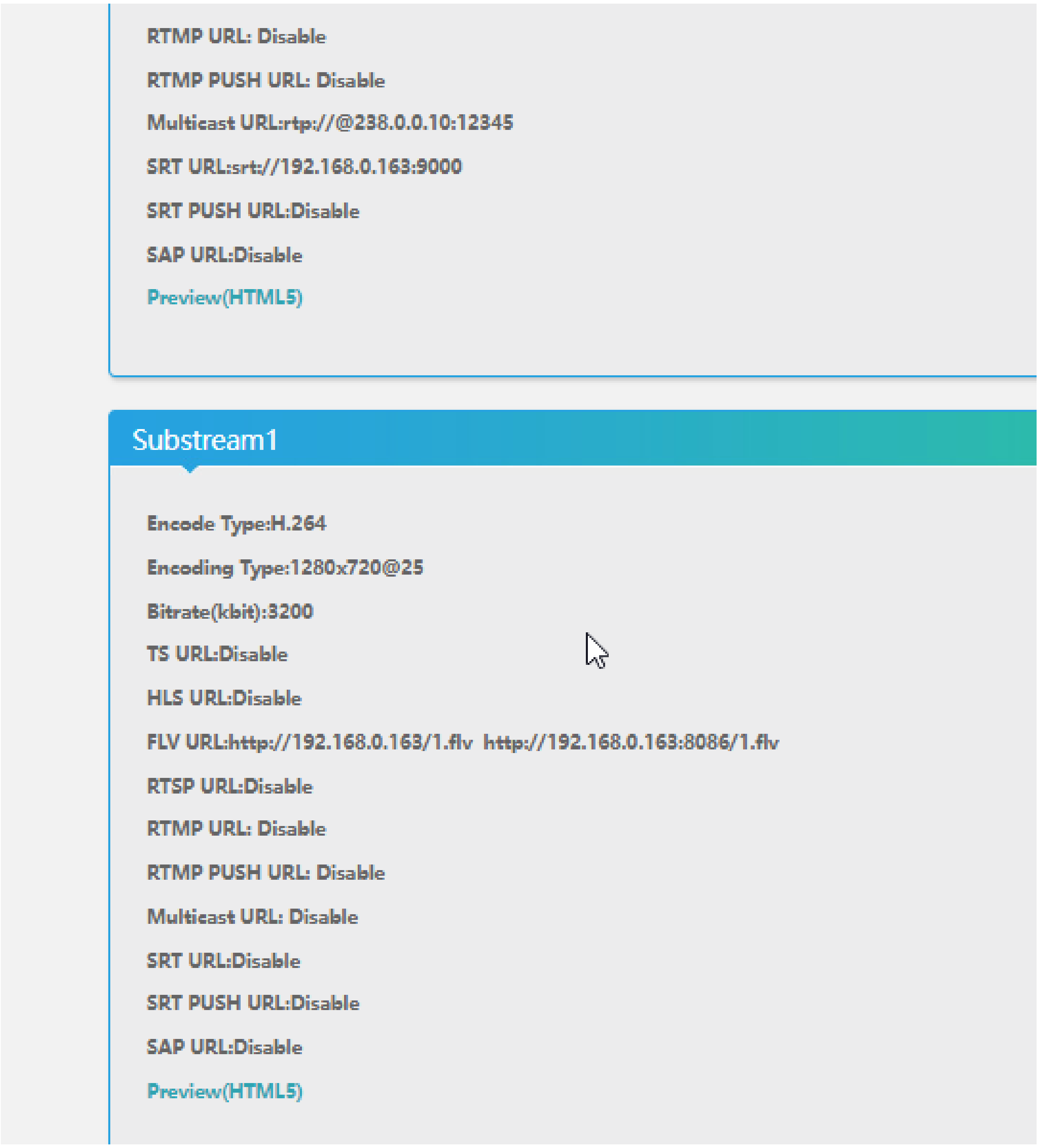


This DOES NOT MEAN to restart your encoder but rather your decoder because you changed encoding settings and the decoder -like VLC- must adapt itself to the new values.

➔ **STATUS Window:**

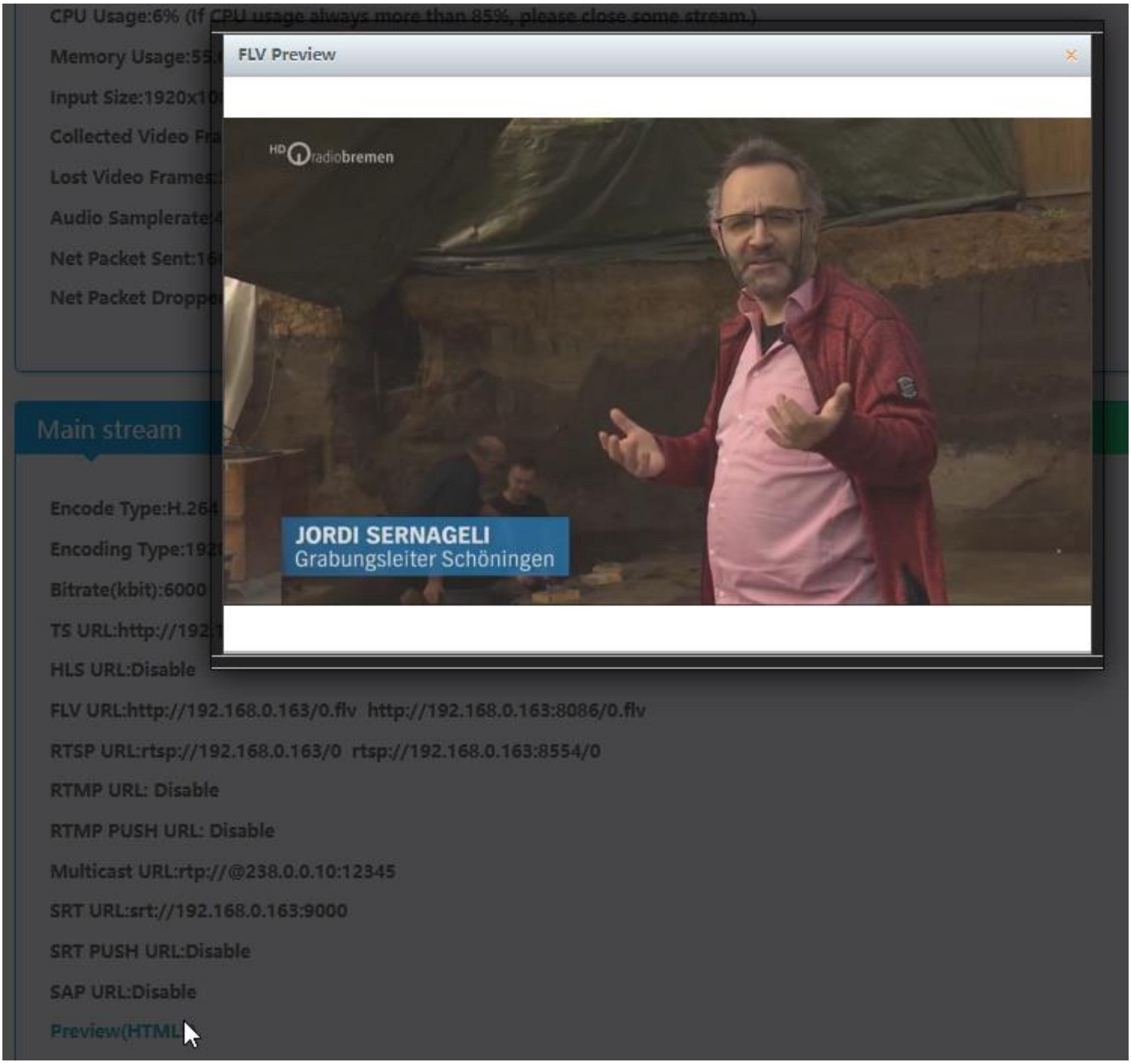


You'll get more information if scrolling down:

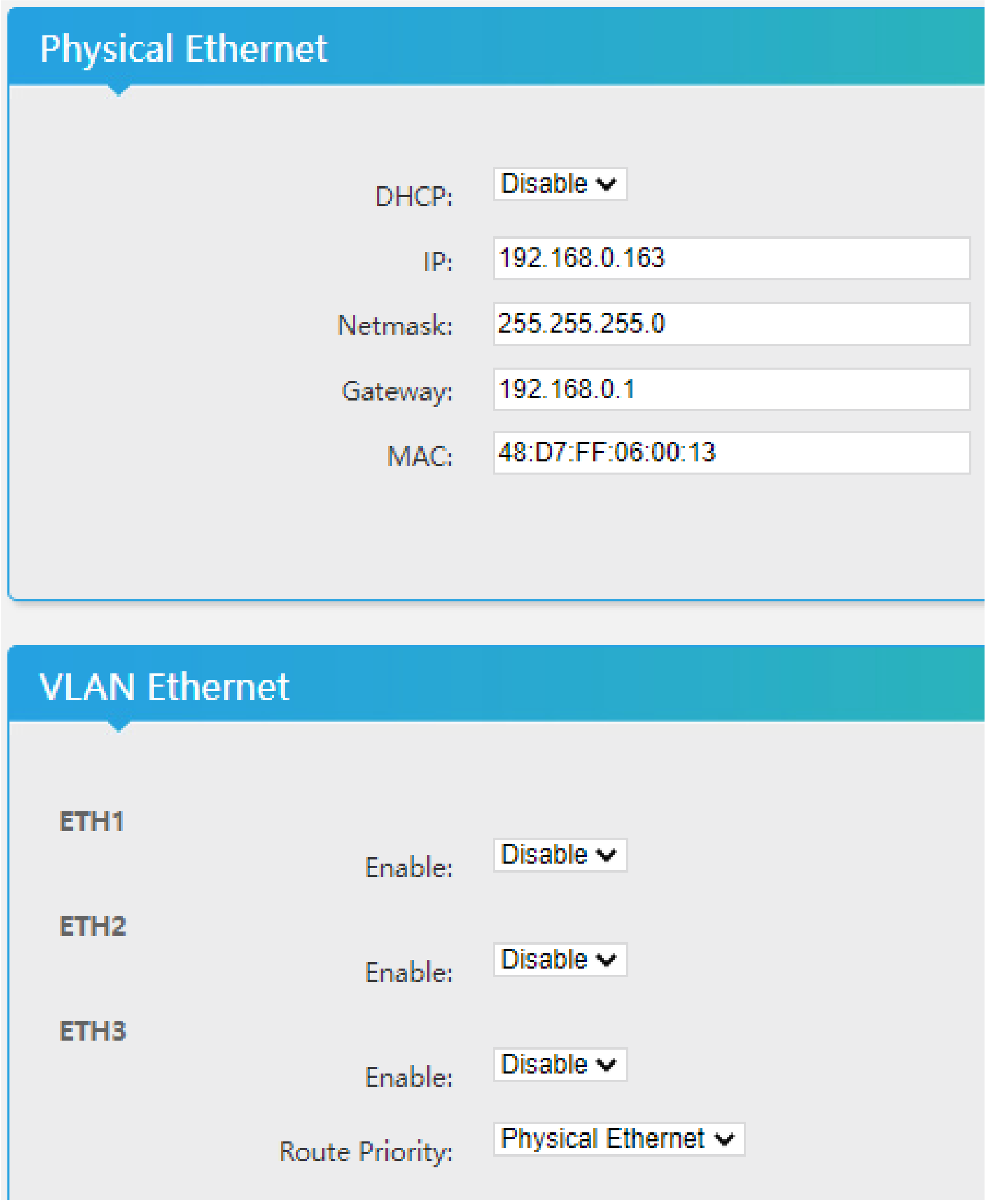


The sub-menus are at the bottom.
Remark: The PREVIEW popup window is not available in all models…

And: FLV stream must be switched **on** to view it:

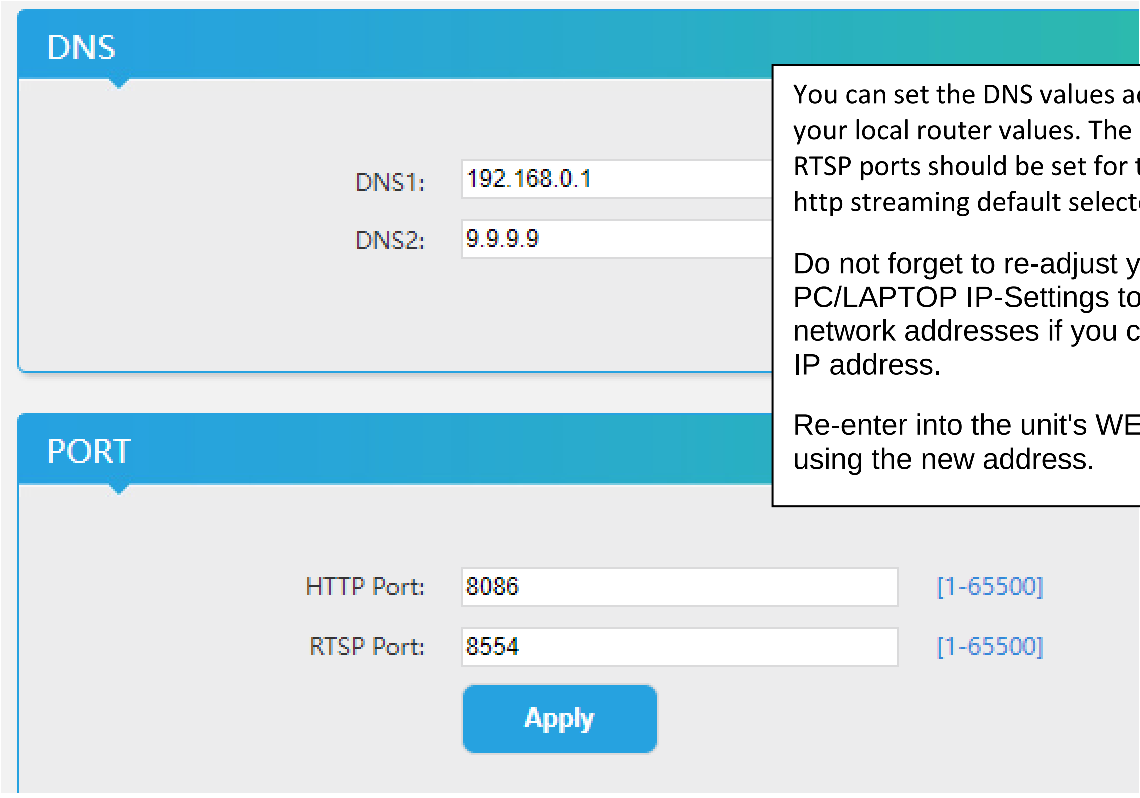

CPU Usage:6% (If CPU usage always more than 85%, please close some stream.)
Memory Usage:55.
Input Size:1920x10
Collected Video Fra
Lost Video Frames:
Audio Samplerate:4
Net Packet Sent:16
Net Packet Droppe
FLV Preview
HD radiobremen
JORDI SERNAGELI
Grabungsleiter Schöningen
Main stream
Encode Type:H.264
Encoding Type:192
Bitrate(kbit):6000
TS URL:http://192.1
HLS URL:Disable
FLV URL:http://192.168.0.163/0.flv http://192.168.0.163:8086/0.flv
RTSP URL:rtsp://192.168.0.163/0 rtsp://192.168.0.163:8554/0
RTMP URL: Disable
RTMP PUSH URL: Disable
Multicast URL:rtp://@238.0.0.10:12345
SRT URL:srt://192.168.0.163:9000
SRT PUSH URL:Disable
SAP URL:Disable
Preview(HTML

Network settings submenu (bottom-menu):



some has VLAN support.

DHCP might not be a good idea because your local network router would handover an IP Address from his pool which you can only get by entering the router itself or use an IP scan-tool.
Remark: DNS settings might not be necessary because the device would not need to use them to translate domains <-> IP addresses.
Please change the settings to your local network values and scroll down to safe it by pressing SET UP:

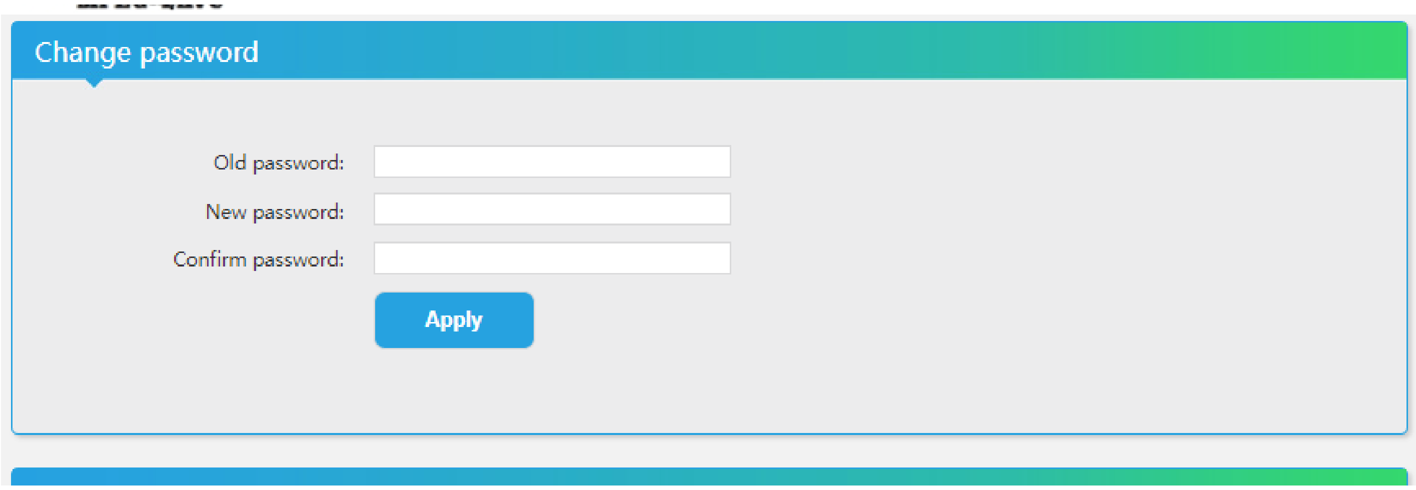


You can set the DNS values according to your local router values. The HTTP and RTSP ports should be set for the unicast / http streaming default selected ports.

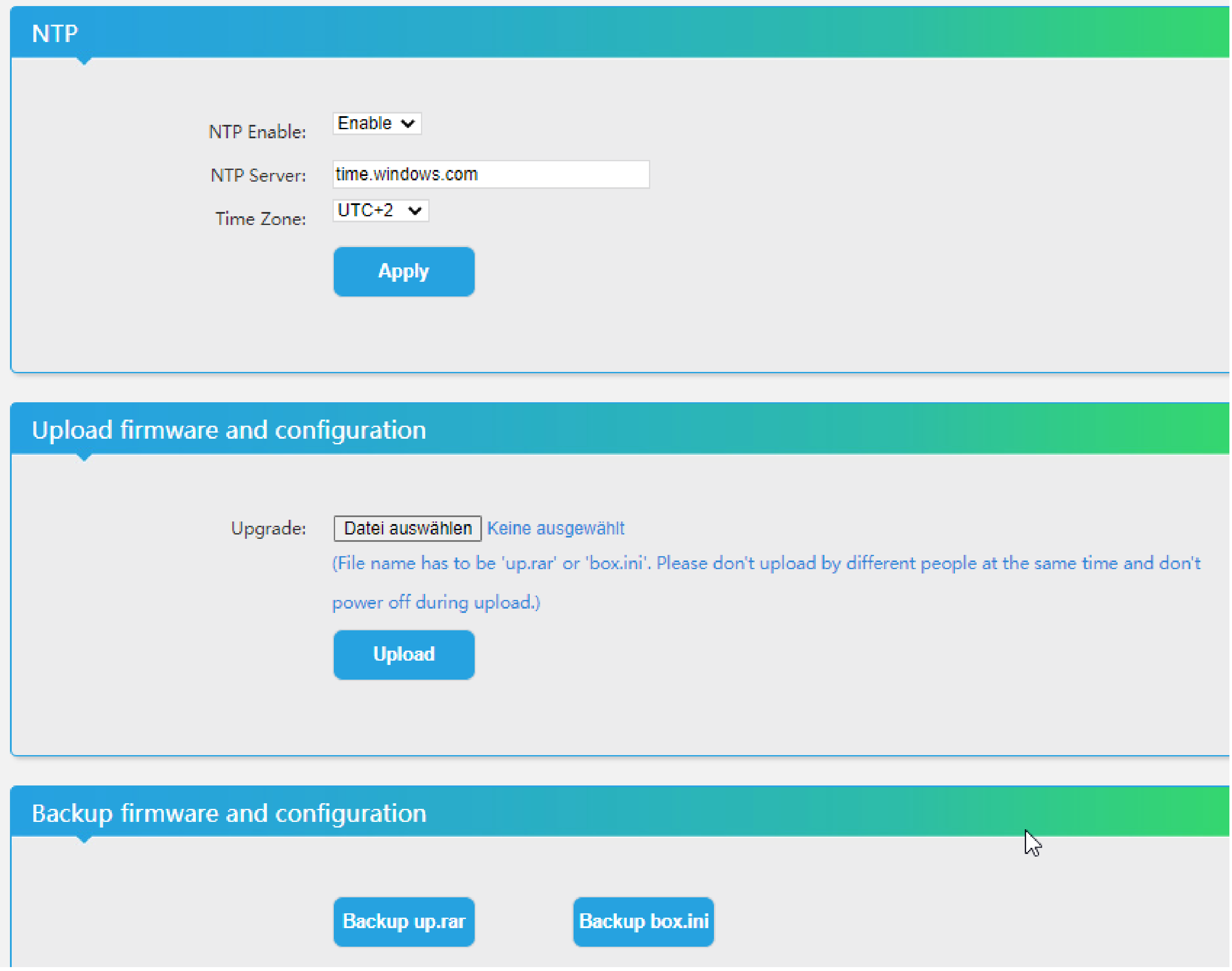
Do not forget to re-adjust your PC/LAPTOP IP-Settings to your local network addresses if you changed the IP address.

Re-enter into the unit's WEB-IF by using the new address.

Enter the SYSTEM Menu

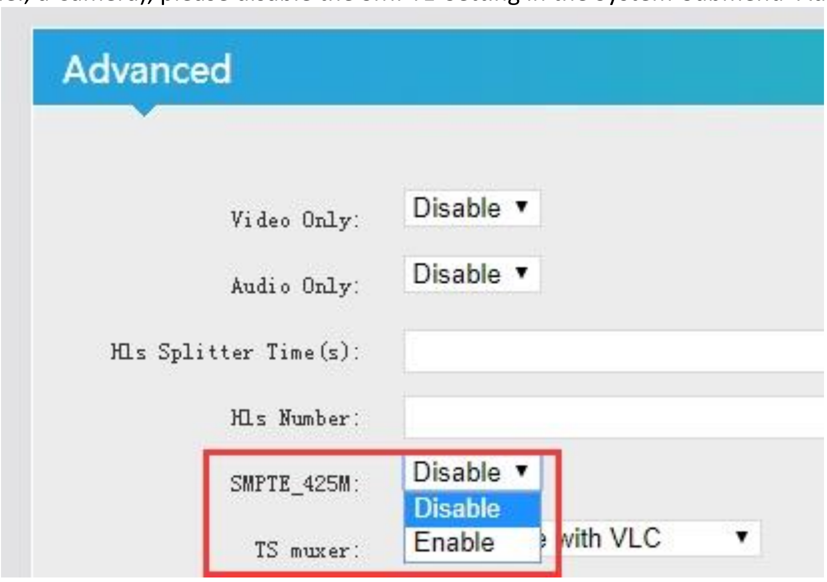


-To reboot your device and to enable the new settings or
-Do a SW-update, set the time-zone for the NTP – fetcher (UTC+1 or 2 = Germany) …
Rem.: Firmware updates will not be published, so please ask us if you have some problems or want to use new features implemented like: (1.1.2019): MJPEG encoding support (in particular models), HLS slicing, RTMPs nginx proxy support, …

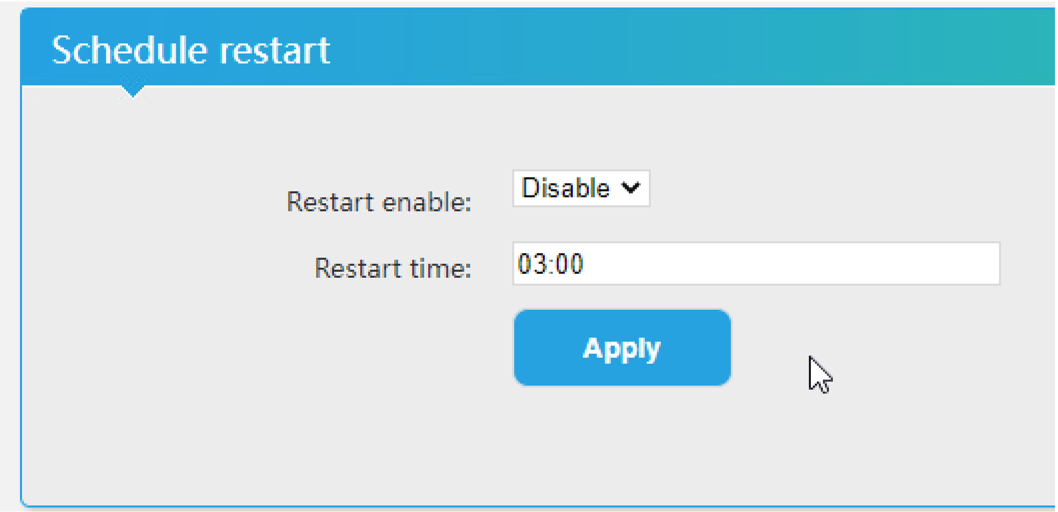


Some hints for SDI-Encoder types:

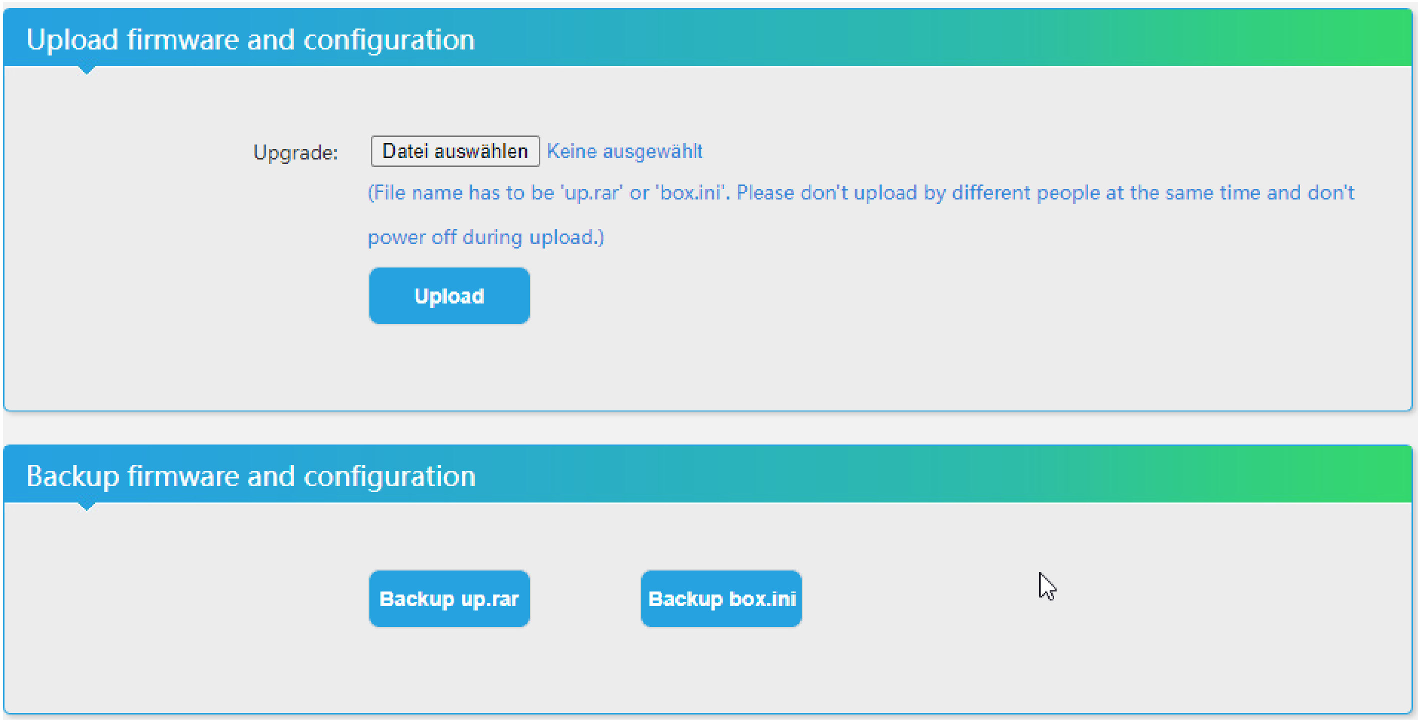
Because the SDI signal has some other values to consider than HDMI Inputs,
If you face problems with the SDI-Inputs detecting and showing only 1080p as Input regardless of the real Input is a 1080i (Interlaced) Video source (i.e., a Camera), please disable the SMPTE-Setting in the System-Submenu 'Advanced' section:



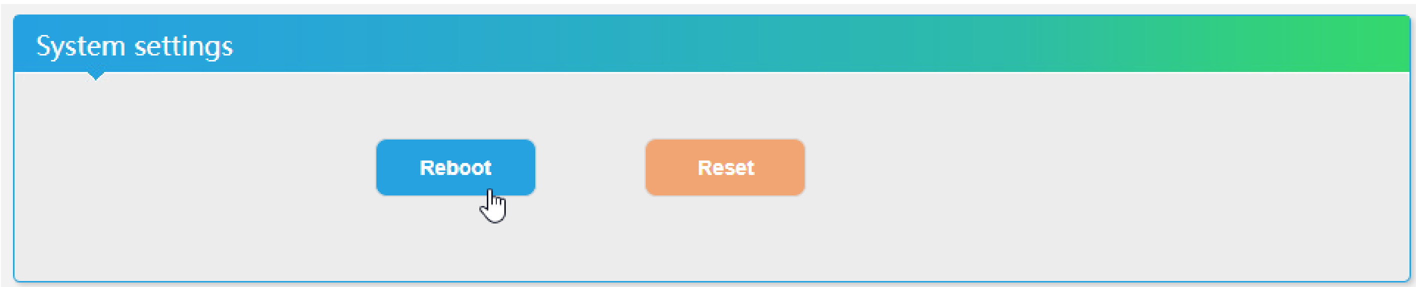
Configure a planned scheduled restart:



Upgrade the Firmware and initialize a reboot:



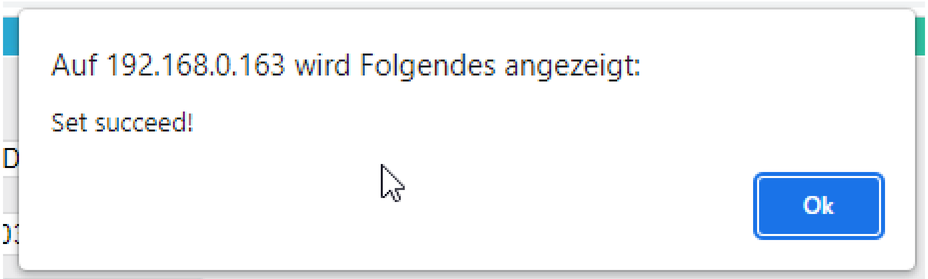
The settings as well as the Firmware can be back-upped and re-uploaded.



The config-settings-file is a Linux based text file named box.ini. Do not modify that by a windows editor except you will use notepad++ (freeware – please google...). Windows has different CR/LF notation than Linux based TXT-files.

Finally, after a firmware update has been uploaded, the unit can be remotely reset to factory defaults or rebooted.

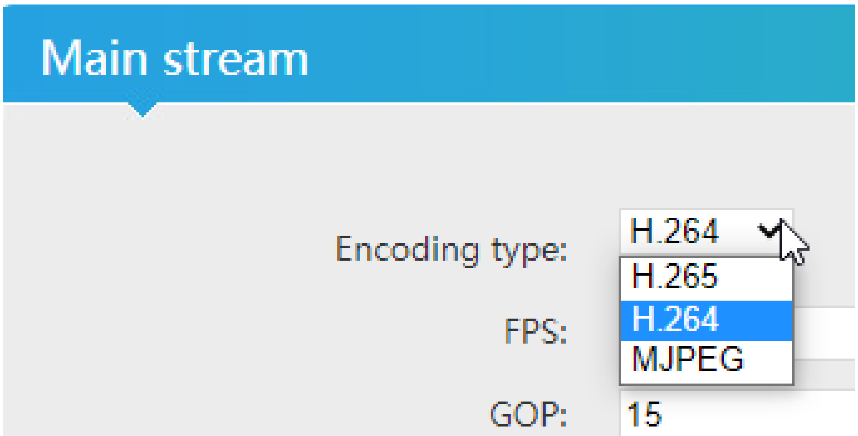
A Pop-Up Message will inform you to confirm the changes you made:



Main stream encoding settings:

We assume, that the user already well knows the relevant terms and abbreviations for Video-Encoding and their technical background.

Choose your codec. (New in 2019: MJPEG support)

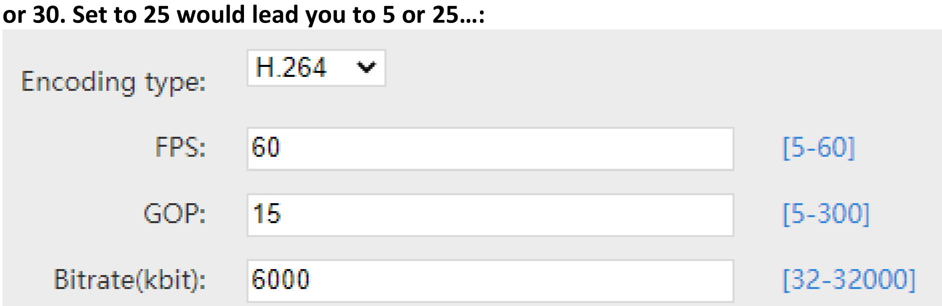


Main - basic video encoding settings:

- Enc type: h.264 & h.265 (is optional dep. on model) or MJPEG (new in 2019)
  Note: h.264 version can only use h.264 codec, h.265 model can do both
- Profile: baseline profile / main profile / high profile (note: h.265 version select main profile)
- Frame rate: 5-60 (If the input resolution is 720i/50,1080i50, the frame rate will be set to 25)
- Bitrate mode: VBR / CBR variable or constant while CBR is the better choice
- Group of pictures: 5-200, shows picture quality, default setting is almost sufficient
- Bitrate: 16-16000 kbit/s or 32-32000 kbit/s dep. on Model (Network bandwidth setting)

Note: You can modify the above parameters without rebooting the encoder
Some encoders like 4K version supporting GOP settings in 4 different modes.

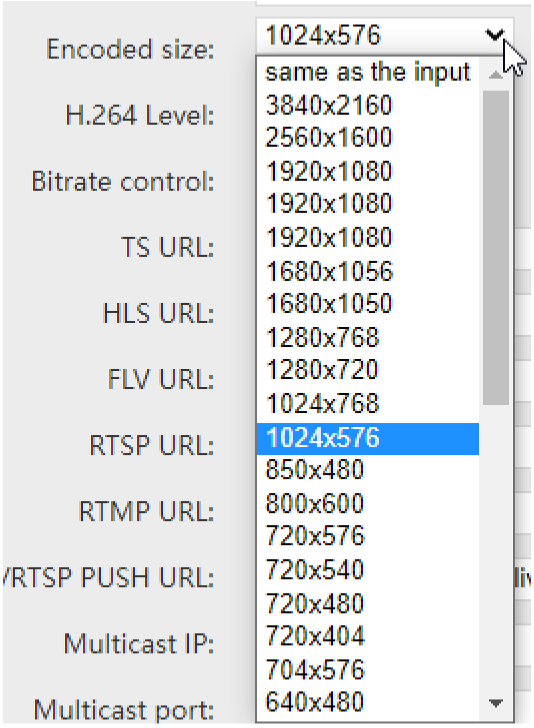
The GOP should be an integer factor / divider from the selected FPS/Hz value. So if you set your fps to 30, it should be 5, 6, 15 or 30. Set to 25 would lead you to 5 or 25…:

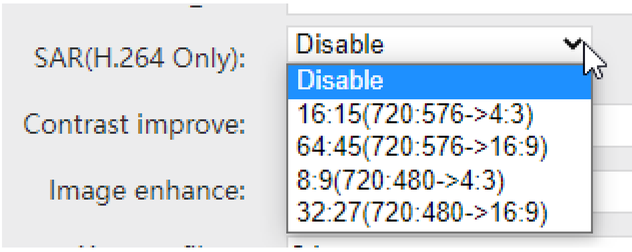


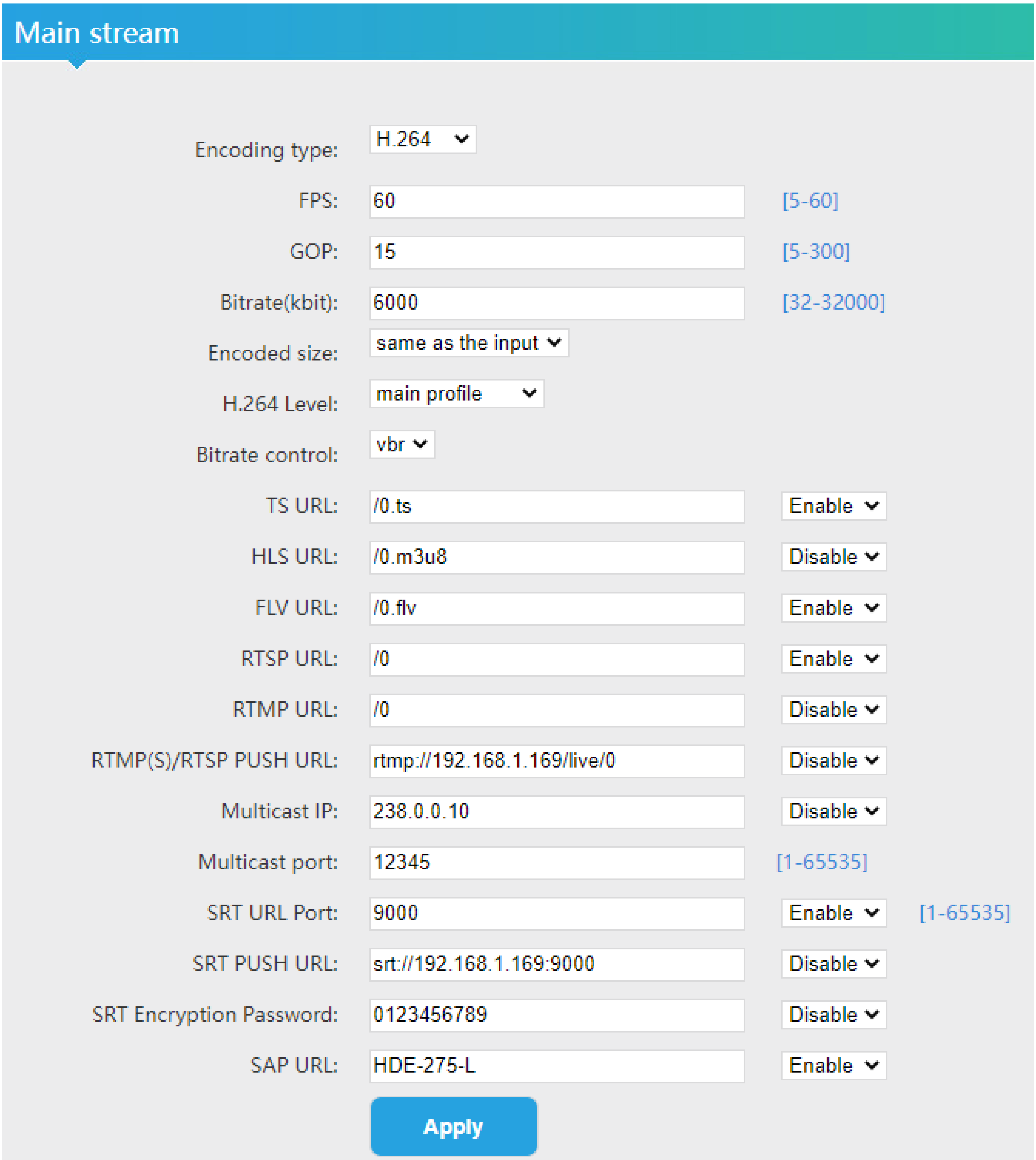
Selecting your output screen size/resolution:

Note: The encoder is *not intent* to use for upscaling purpose i.e. 720 -> 1080. It always depends on the Input resolution and size.

Downscaling is possible but you should calculate also the SAR settings than:


Encoded size:
H.264 Level:
Bitrate control:
TS URL:
HLS URL:
FLV URL:
RTSP URL:
RTMP URL:
/RTSP PUSH URL:
Multicast IP:
Multicast port:
1024x576
same as the input
3840x2160
2560x1600
1920x1080
1920x1080
1920x1080
1680x1056
1680x1050
1280x768
1280x720
1024x768
1024x576
850x480
800x600
720x576
720x540
720x480
720x404
704x576
640x480


SAR(H.264 Only):
Contrast improve:
Image enhance:
Disable
Disable
16:15(720:576->4:3)
64:45(720:576->16:9)
8:9(720:480->4:3)
32:27(720:480->16:9)



Main - Stream encoding & protocol settings should be crosschecked with the SYSTEM-Advanced settings because you can set here the common defaults for all stream outputs:

HTTP:	/main enable/disable
HTTP port:	1-65535 optional
RTSP:	/main enable/disable
RTSP port:	1-65535 optional
Multicast IP:	232.255.42.42 disable/RTP/UDP optional
Multicast port:	1-65535 optional
RTMP server IP:	can be set according your streaming media server values
RTMP server port:	1-65535 optional
RTMP app name:	can be set by yourself
RTMP stream name:	can be set by yourself
RMTP user name:	User for your server

RMTP password name: and Password for your server

ONVIF : enable/disable (IP-Camera protocol support) -> **Needs RTSP** stream to ON

REM: maybe better to use RTP instead of UDP should to be selected in the SYSTEM Menu …

Note: ONVIF Settings depending on SW and device model types

Almost all our Encoder models are coming with ONVIF support and can be used with following protocols: "ONVIF S, "ONVIF C" or "ONVIF G"

Example: HDMI-encoder ONVIF worked with Genetec VMS like:

ONVIF Device Manager is a Network Video Client (NVC) to manage Network Video Transmitters (NVT), Network Video Storage (NVS) and Network Video Analytics (NVA) devices. Implements Discovery, Device, Media, Imaging, Analytics, Events and PTZ services. Written in C# and uses ffmpeg for media decoding.
Downloading:
https://sourceforge.net/projects/onvifdm/
English User Guide
https://wiki.allprojects.info/display/ODMDOC/ONVIF+Device+Manager.+Installation+and+User+Guide
https://wiki.2n.com/hip/inte/latest/en/8-vms/onvif-device-manager
https://www.happytimesoft.com/products/onvif-server/index.html

I would like to clarify the following:

The Encoder sent to Genetec HQ for integration is a
[redacted] (HDMI)
This has been tested in our lab and works properly.

The Encoders used in the CESAC project are
[redacted] (DB15 Analog input)
This is the one we have not been able to stream video from.

Rabindranath Parra
Professional Services Manager - Latin and South America

Genetec

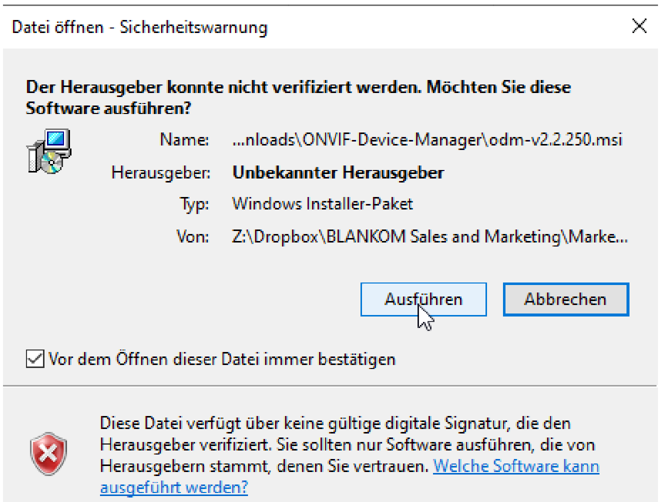
Video Surveillance | Access Control | License Plate Recognition

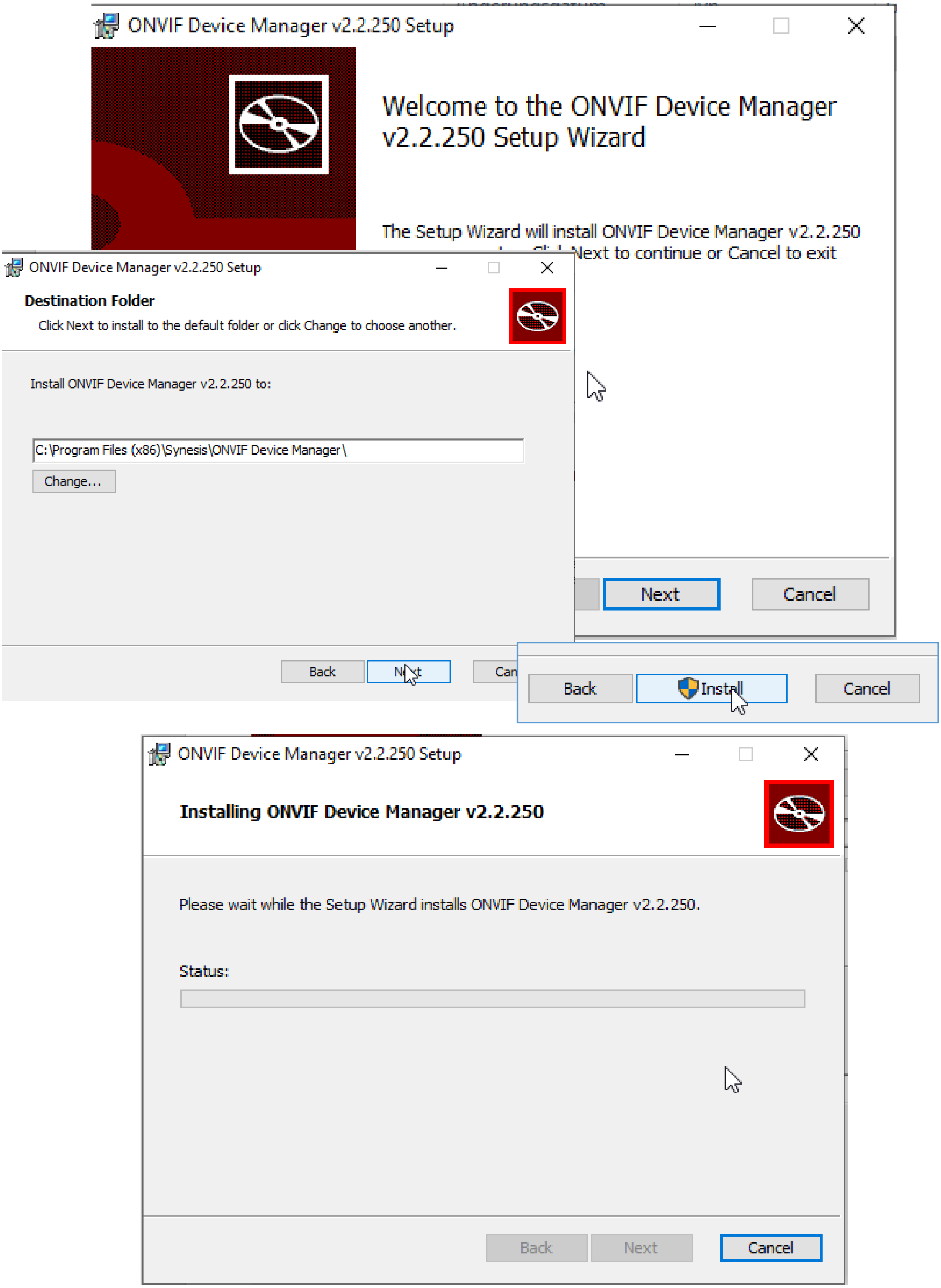
P: +1-514-332-4000 X 6781 | M: +52 1 (443) 273-2460 | rparra@genetec.com
2280, Alfred-Nobel Blvd, suite 400, Montreal, QC, H4S 2A4, Canada

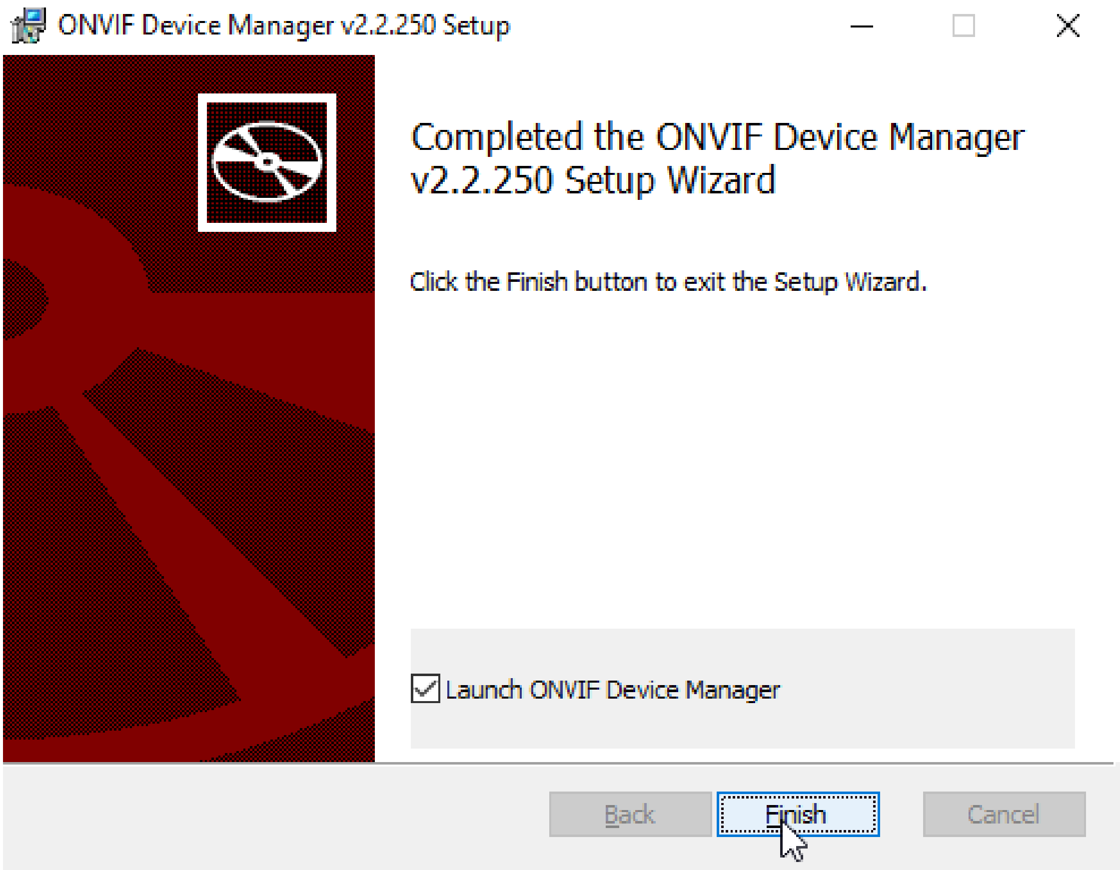
Built to evolve: www.genetec.com/dna

Download ONVIF-Device-manager: from a link in our Web – Section 'DOWNLOADS'
odm-v2.2.250.msi - for Windows OS.

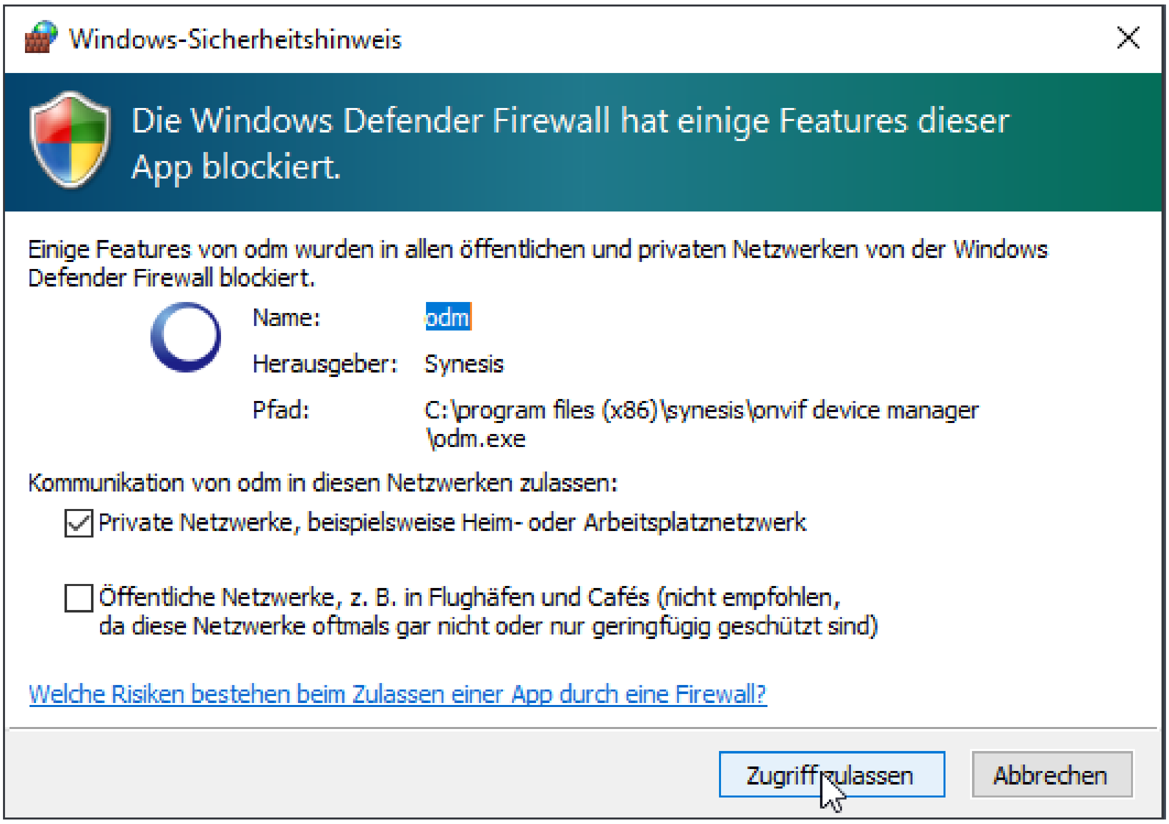
Installation as Admin maybe: but it's not a signed MS Application:



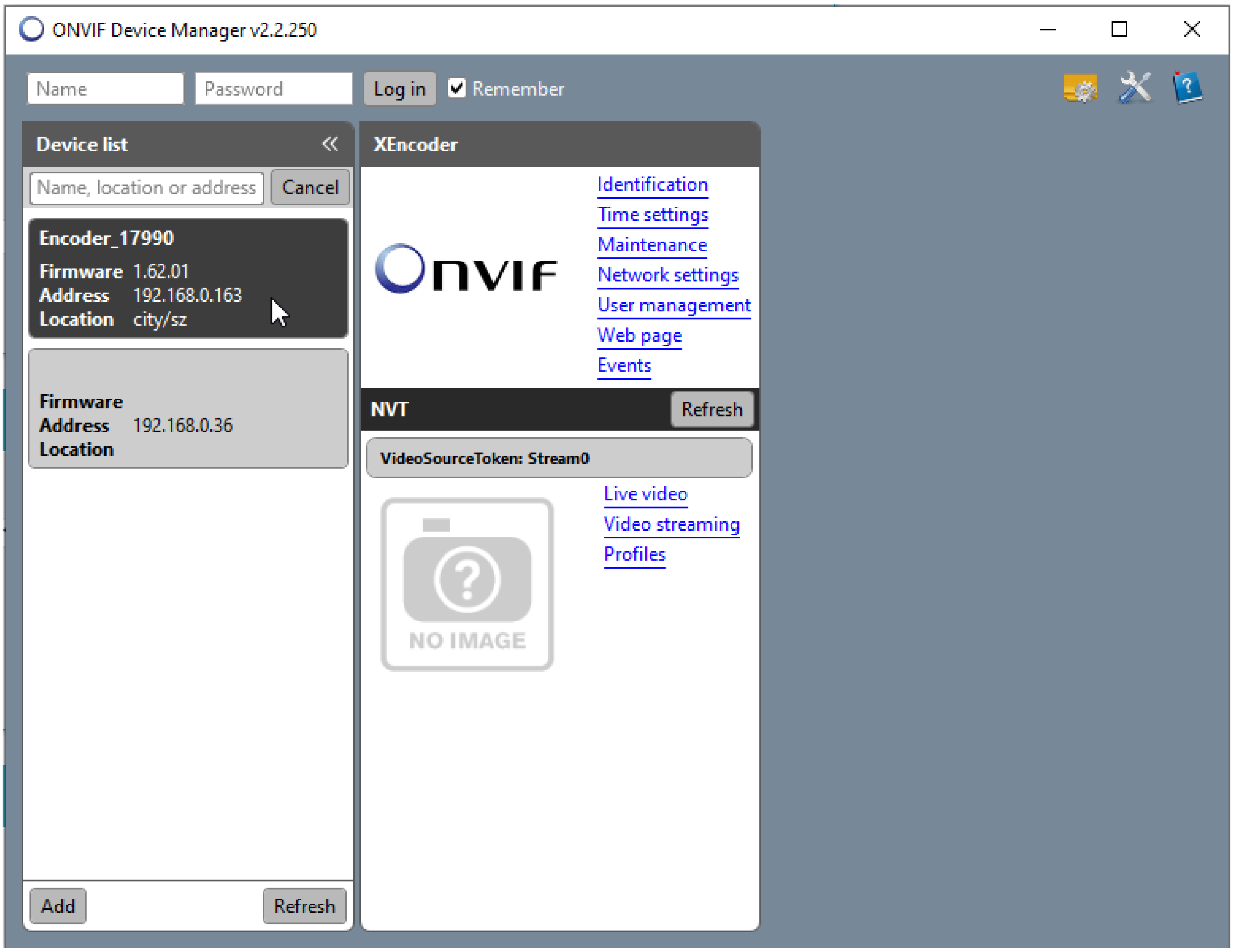

ONVIF Device Manager v2.2.250 Setup
Welcome to the ONVIF Device Manager v2.2.250 Setup Wizard
The Setup Wizard will install ONVIF Device Manager v2.2.250
Next to continue or Cancel to exit
Next
Cancel
ONVIF Device Manager v2.2.250 Setup
Destination Folder
Click Next to install to the default folder or click Change to choose another.
Install ONVIF Device Manager v2.2.250 to:
C:\Program Files (x86)\Synesis\ONVIF Device Manager\
Change...
Back
Back
Install
Cancel
ONVIF Device Manager v2.2.250 Setup
Installing ONVIF Device Manager v2.2.250
Please wait while the Setup Wizard installs ONVIF Device Manager v2.2.250.
Status:
Back
Next
Cancel



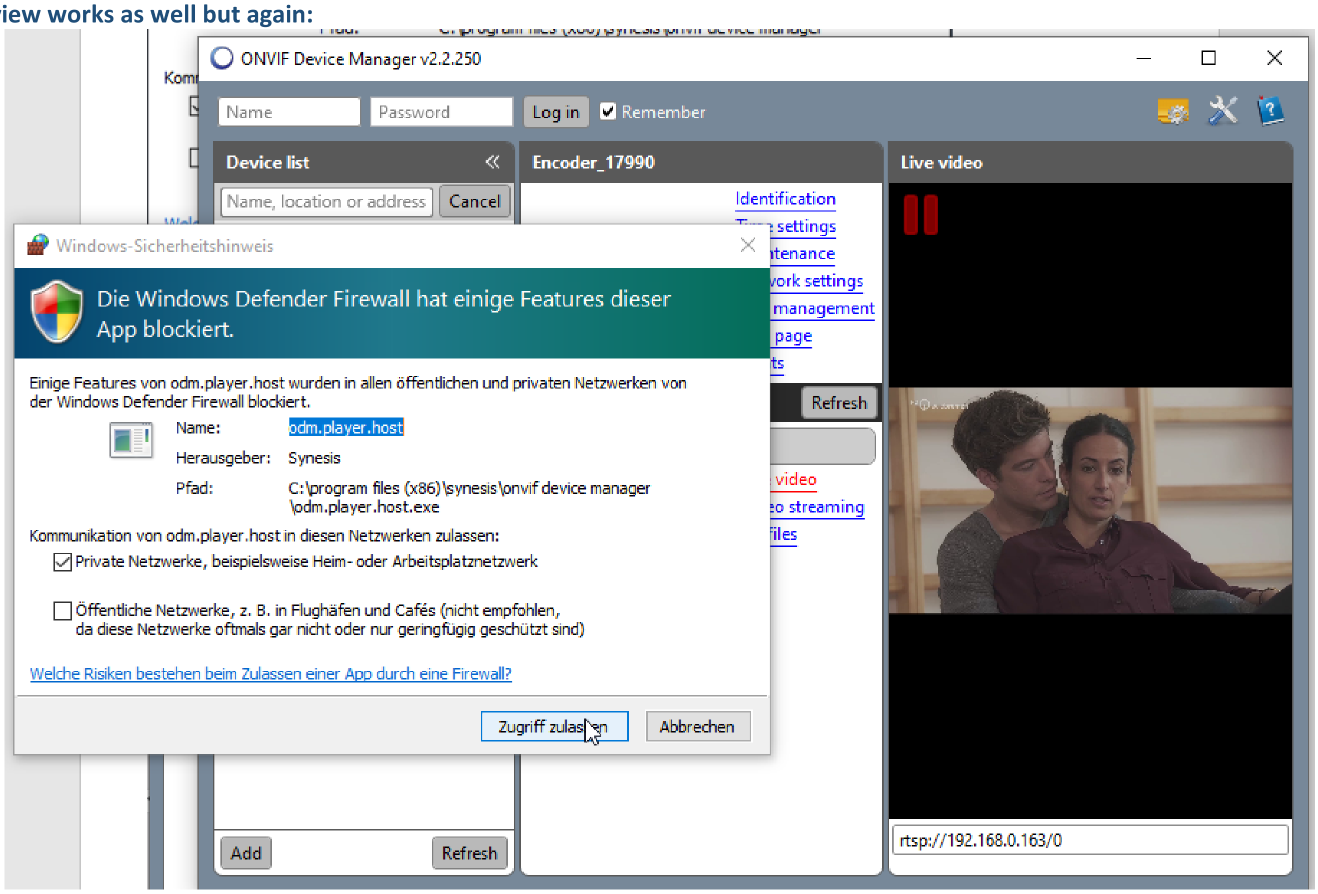
It asks for admin rights and Firewall access:

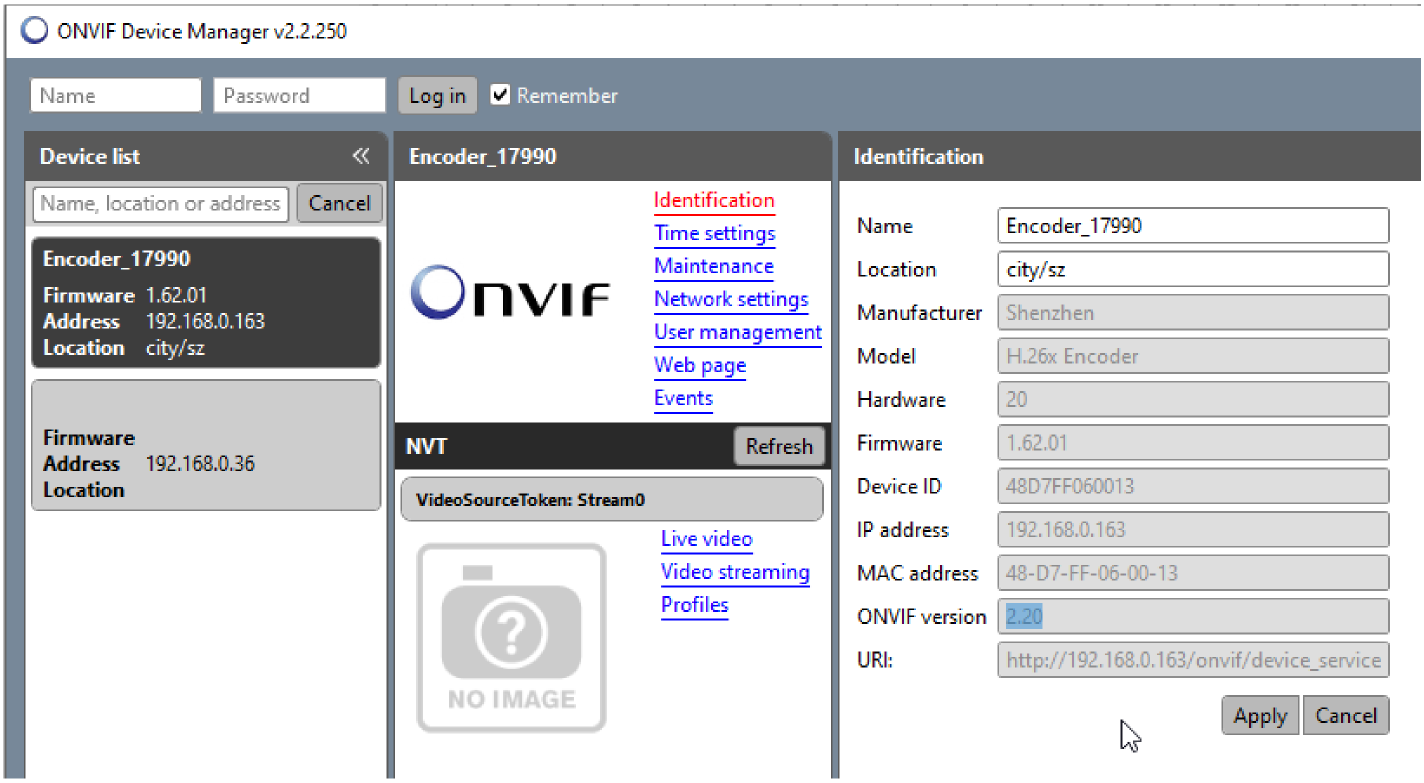


E' Voila our encoder has been detected:



Preview works as well but again:

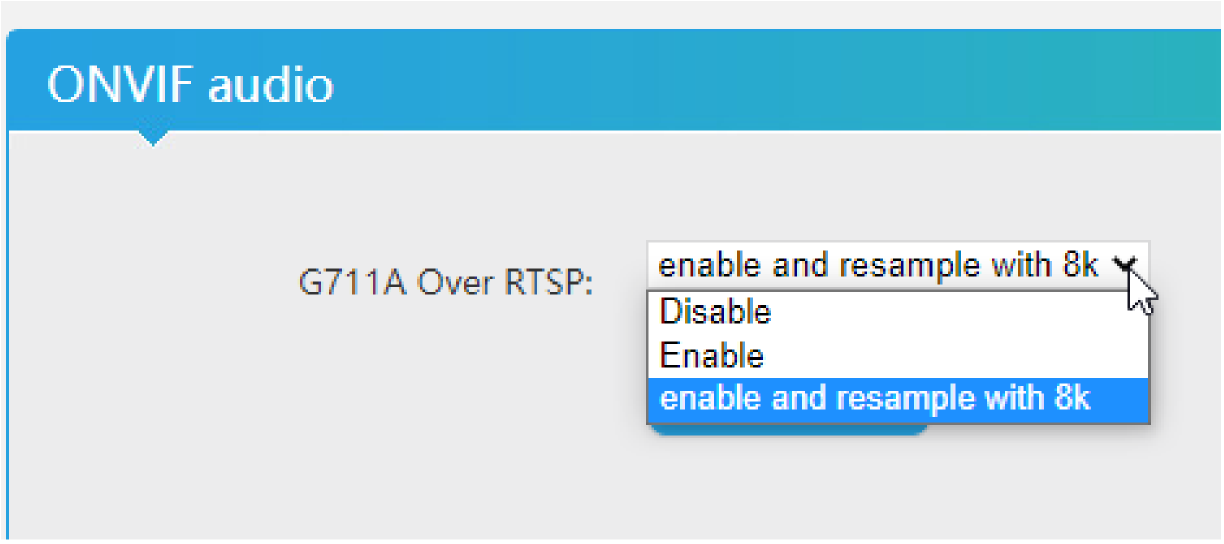


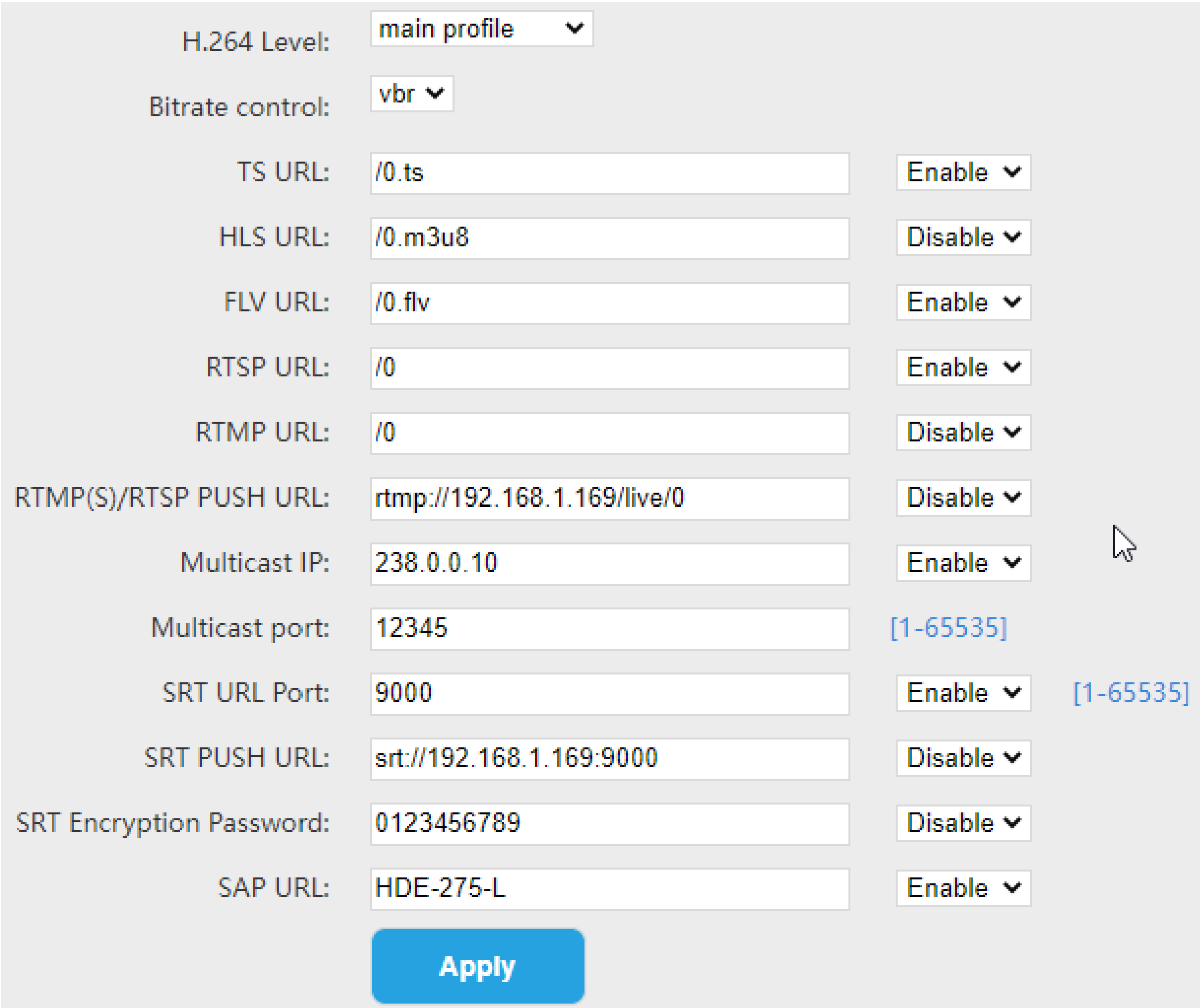


ONVIF Version in Display is 2.20… so this manager software is a little older ….. Russian ….
You can login by ONVIF:

We do not go any further with the ODM now, please google and try to find some more info’s.

Note: We recommend to choose or enable not all streams and protocols at the same time.
RTMP(s) is almost not working with h.265 HEVC-Codec because Adobe hasn't included this Codec into its list of valid codecs yet. Maybe Adobe and also Apple will be more open in the future or when you read this, they have integrated it.
For AC3 Stereo, RTSP do not support it, so, when you enable AC3, RTSP will use the G711A.
If you choose AC3, you can't disable the G711A audio for ONVIF:





Note: For TS URL, HLS URL, RTSP URL, the Network IP address of the device is the Unicast distribution address. You do not need to fill in this address by yourself because it is already related to the RJ45-Ethernet network IP. -264 version have Fast-Ethernet only.

Multicast IP addresses and ports need to be considered in the IANNA recommended range to avoid conflicts within your local network and router/switches:
https://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

TS once pack:

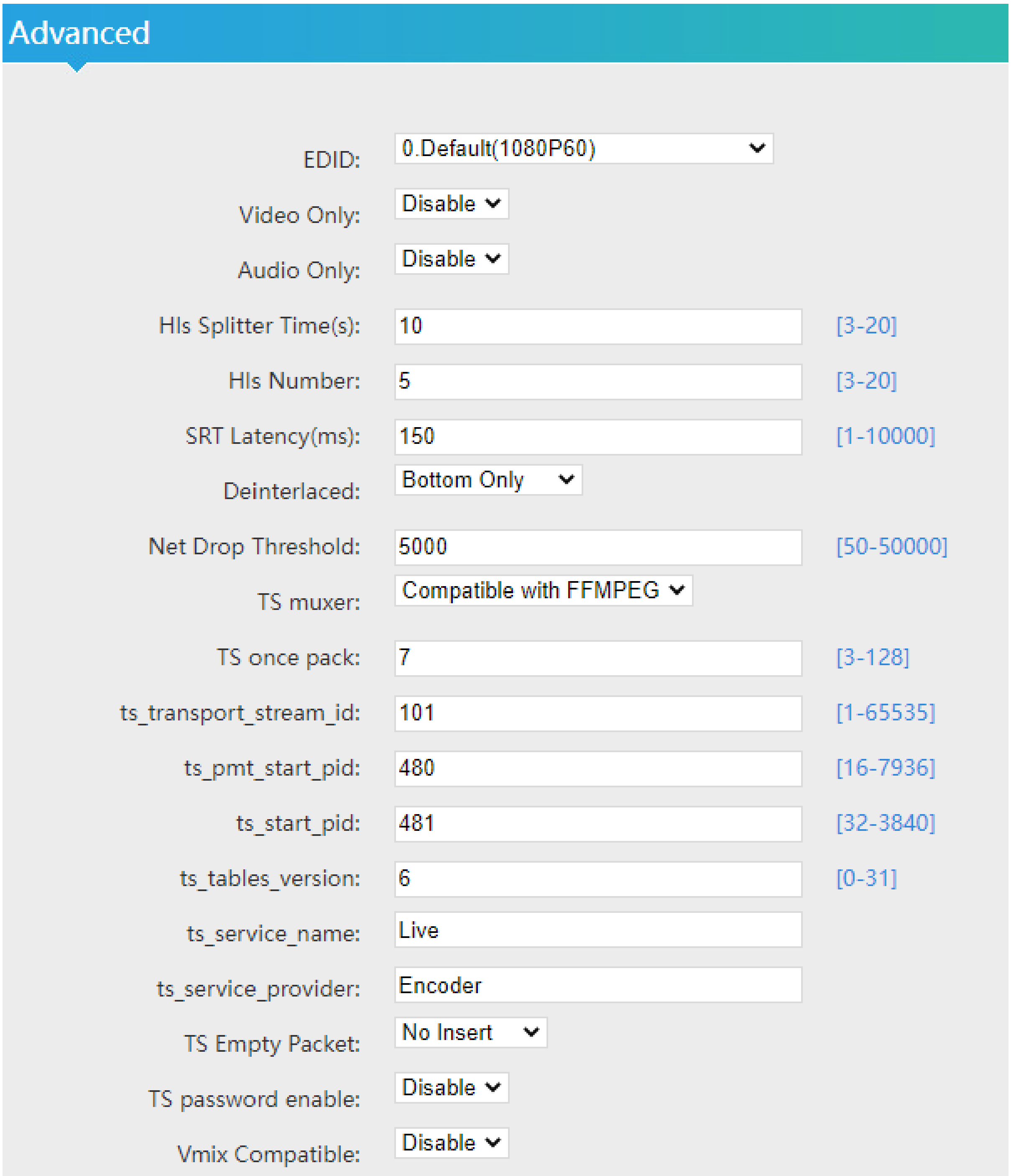
is the packet size send by the encoder streamer.

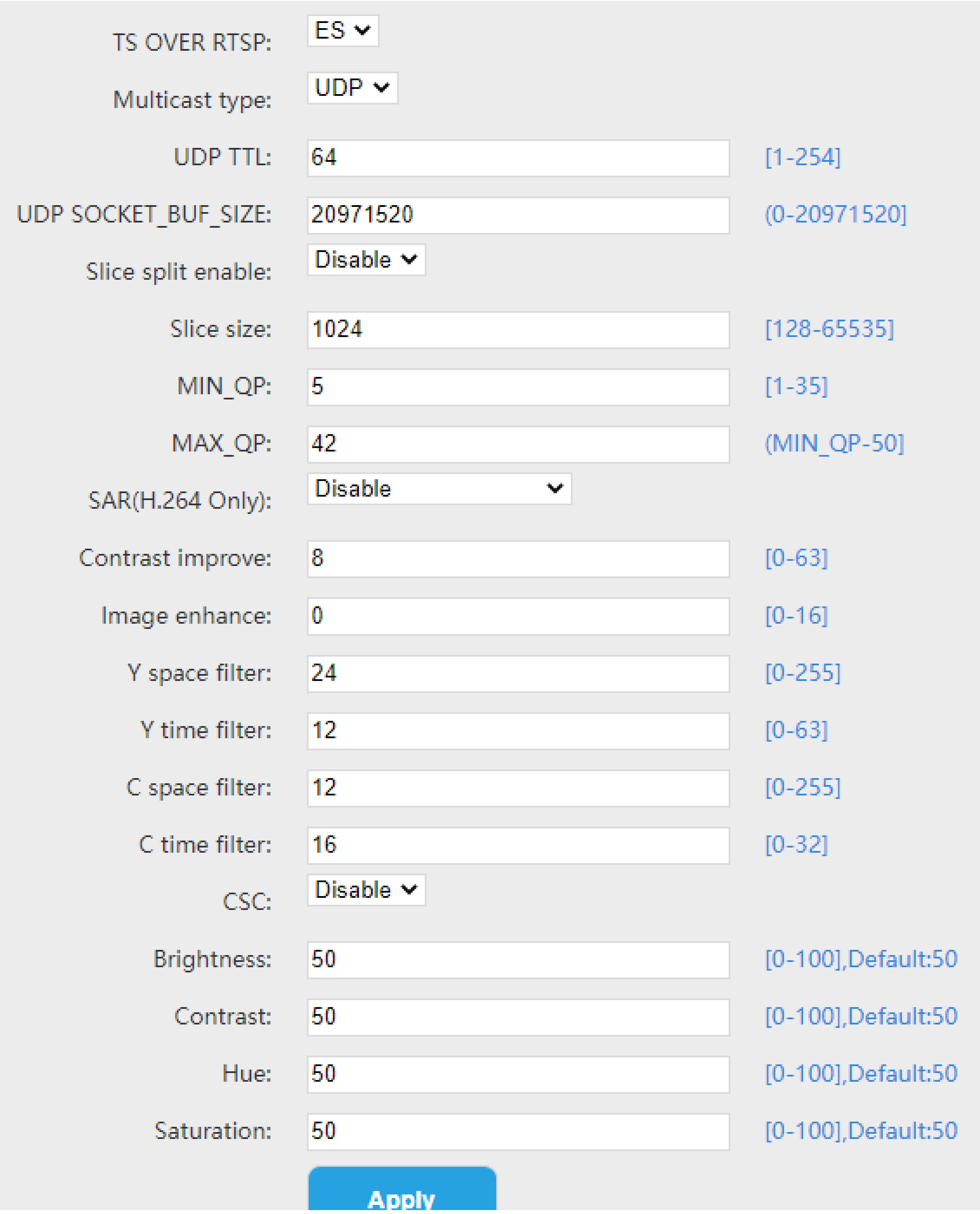
Usually, we suggest to set it to 7.

For example, for UDP:

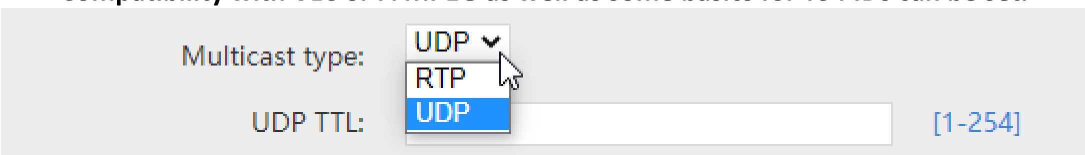
1 packet is 188, 7 pack size will be 188*7+46=1362

SYSTEM Settings:



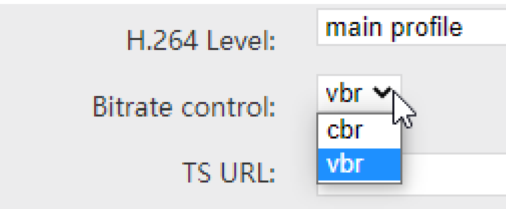


Compatibility with VLC or FFMPEG as well as some basics for TS PIDs can be set.

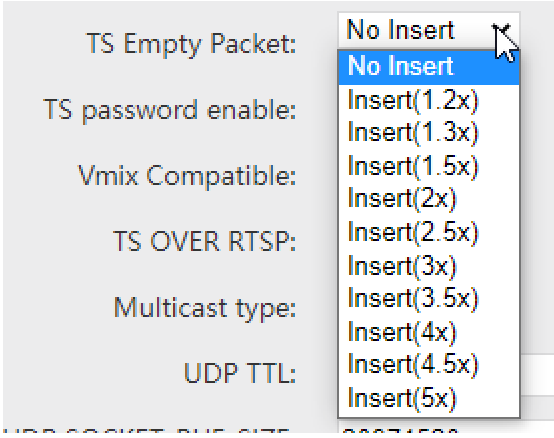


Choose either UDP or RTP for all Multicast outputs.

TS empty packets will insert PID8191dec Zero packets as a factor.
Works in combination with:

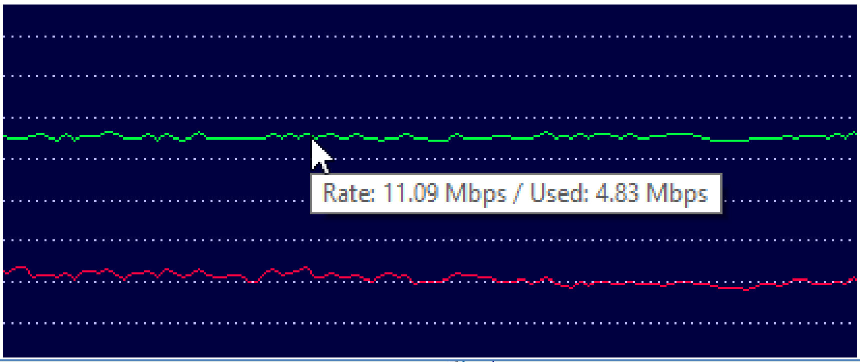


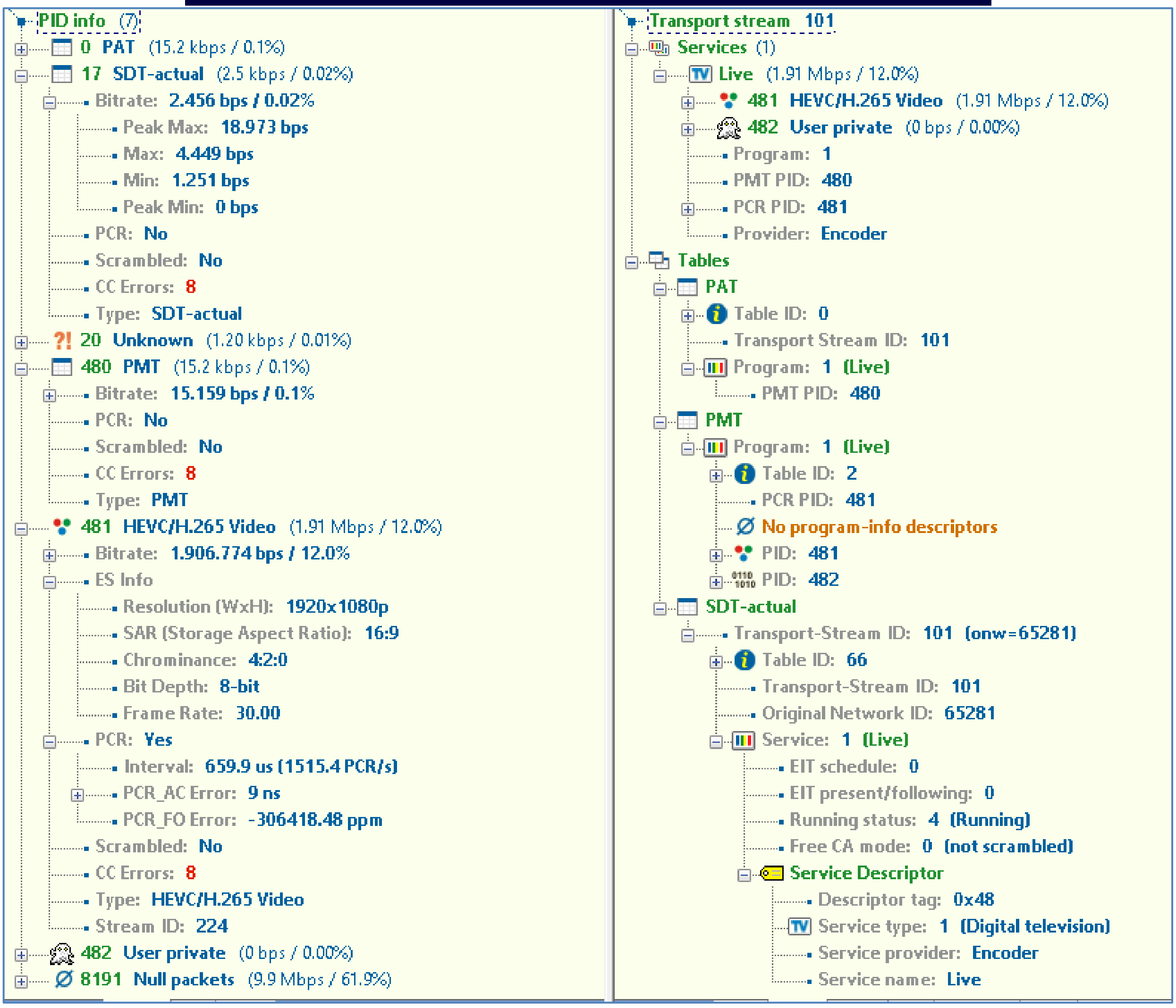
So, no real TS output with a very stable and constant DVB-conform CBR stream will be created:



Do not mess up with CBR encoding parameters and CBR DVB-Zero Packet insertion.
Both abbreviations describing Constant Bit Rate CBR but are 2 different issues.

But will be needed in systems where finally a CBR is used and re-multiplexers are connected. Here the (1.3x) would be the best choice:





PID 8191 dec = Zero-packets injected

Some Encoder models have 'STRICT CBR' as an additional configuration which would improve the Null-Packet insertion.

Serial to TCP – if implemented in the SoC Encoder – Model:

works in combination with
an integrated remserial-1.4 function:

Remserial

The remserial program acts as a communications bridge between a TCP/IP network port and a Linux device such as a serial port. Any character-oriented Linux /dev device will work.

The program can also use pseudo-ttys as the device. A pseudo-tty is like a serial port in that it has a /dev entry that can be opened by a program that expects a serial port device, except that instead of belonging to a physical serial device, the data can be intercepted by another program. The remserial program uses this to connect a network port to the "master" (programming) side of the pseudo-tty allowing the device driver (slave) side to be used by some program expecting a serial port. See example 3 below for details.

The program can operate as a server accepting network connections from other machines, or as a client, connecting to remote machine that is running the remserial program or some other program that accepts a raw network connection. The network connection passes data as-is, there is no control protocol over the network socket.

Multiple copies of the program can run on the same computer at the same time assuming each is using a different network port and device.

Some examples:

1) Give access to a RS232 device over a network.

The computer with the serial port connected to the device (such as a data aquisition device) runs the remserial program:

```
remserial -d -p 23000 -s "9600 raw" /dev/ttyS0 &
```

This starts the program in daemon mode so that it runs in the background, it waits for connections on port 23000 and sets up the serial port /dev/ttyS0 at 9600 baud. Network connections to port 23000 from any machine can then read and write to the device attached to the serial port.

This can be started from /etc/rc.local or as an entry in /etc/inittab or set up as a system service with a file in /etc/rc.init/.

2) Connect an RS232 device to a specified server.

The computer with the serial port connected to the device (such as a data aquisition device) runs the remserial program:

```
remserial -d -r server-name -p 23000 -s "9600 raw" /dev/ttyS0 &
```

This would be used with case number 1 above creating an end-to-end serial port connection. What goes in the serial port on one machine would come out the serial port of the other machine. The ports could be running at different baud rates or other serial port settings.

3) Connect a Linux program that needs a serial port to a remote serial port.

Some programs are written to communicate directly with a serial port such as some data aquisition programs. The remserial program can use pseudo-ttys to fool the program into thinking that it is talking to a real serial port on the local machine:

```
remserial -d -r server-name -p 23000 -l /dev/remserial1 /dev/ptmx &
```

This creates a file called /dev/remserial1 which can be used by the data aquisition application as its serial port. Any data sent or received is passed to the remote server-name on port 23000 where a computer configured in case number 1 above passes it to a real serial port.

The remserial program uses the special pseudo-tty master device /dev/ptmx (see man ptmx) which creates a slave device that looks like a normal serial port named /dev/pts/something. Unfortunately, the actual device name created isn't consistent, so the remserial program creates a symbol link from the device name specified with the -l option to the /dev/pts/ name that was created allowing the other application to be configured with a consistent device name.

4) Server farm console control.

Assuming multiple Linux servers (such as web servers) are set up to have a serial port as their console instead of a monitor/keyboard, their serial

ports could be connected to a control server using a multi-port serial board.
On the control server, a copy of remserial is run for each server:

```
remserial -d -p 23000 -s "115200 raw" /dev/ttyS0 &
remserial -d -p 23001 -s "115200 raw" /dev/ttyS1 &
remserial -d -p 23002 -s "115200 raw" /dev/ttyS2 &
remserial -d -p 23003 -s "115200 raw" /dev/ttyS3 &
etc.
```

From any computer on the local network, use a telnet program to connect to the control server on the appropriate port:

```
telnet control-server-name 23002
```

This would connect through the associated serial port to the desired server's console. This example would then give the user console access to the 3rd server.

Careful scripting such as using the Linux "expect" program could allow batches of commands to be run on each server.

Other Linux program useful with remserial

- nc - The netcat program is similar to remserial except that it creates connections between network ports and command line standard input and output.

  For example, with case number 1 above, the following command run on another computer will send the contents of the named file out the serial port used by the remserial program:

  nc server-name 23000 <file-name

  Similarily, the following command will store incoming serial data in a file until the program is manually interrupted:

  nc server-name 23000 >file-name

- telnet - The telnet program is normally used to log into a remote computer, but when used with network ports other than number 23, it operates in a raw data mode.

  For example, with case number 1 above, the following command will allow the user of the telnet program to see incoming serial port data and type data on the keyboard to send to the serial port:

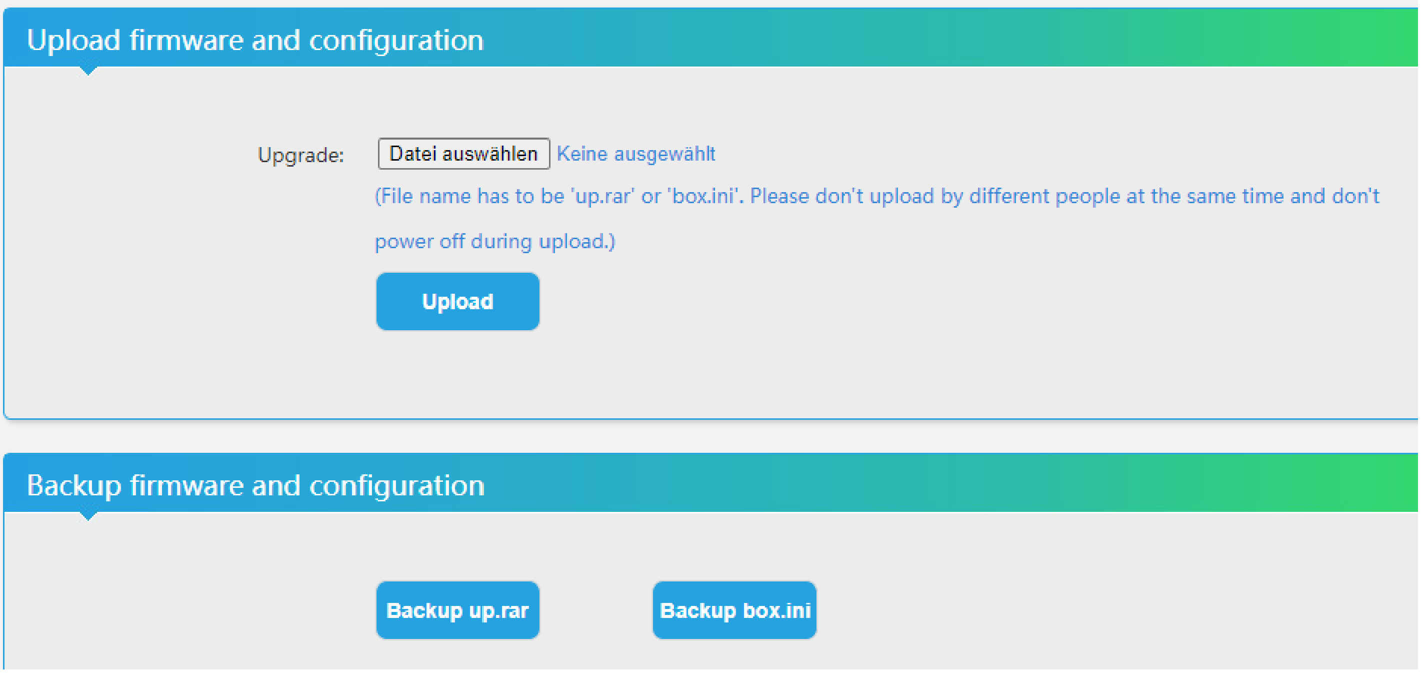
  telnet server-name 23000

  This is ideal for controlling the device connected to the serial port if it has some sort of command line interface usable over the serial port.

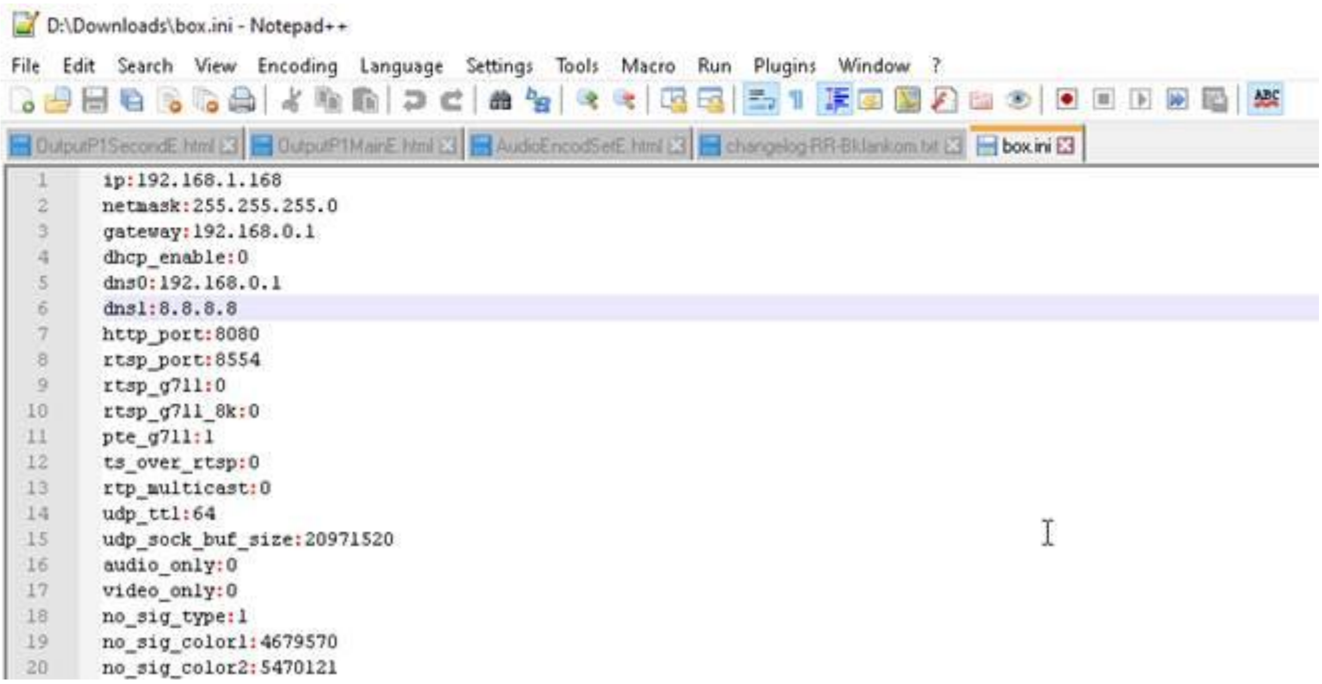
remserial Usage:
----

remserial [-r machinename] [-p netport] [-s "stty params"] device

```
-r machinename          The remote machine name to connect to.  If not
                        specified, then this is the server side.
-p netport              Specifiy IP port# (default 23000)
-s "stty params"        If serial port, specify stty parameters, see man stty
-d                      Run as daemon programs
-x debuglevel           Set debug level, 0 is default, 1,2 give more info
-l linkname             If the device is /dev/ptmx, creates a symbolic link
                        to the corresponding slave pseudo-tty so that another
                        application has a static device name to use.
-m max-connections      Maximum number of simultaneous client connections to allow
device                  Character oriented device node such as /dev/ttyS0.
```

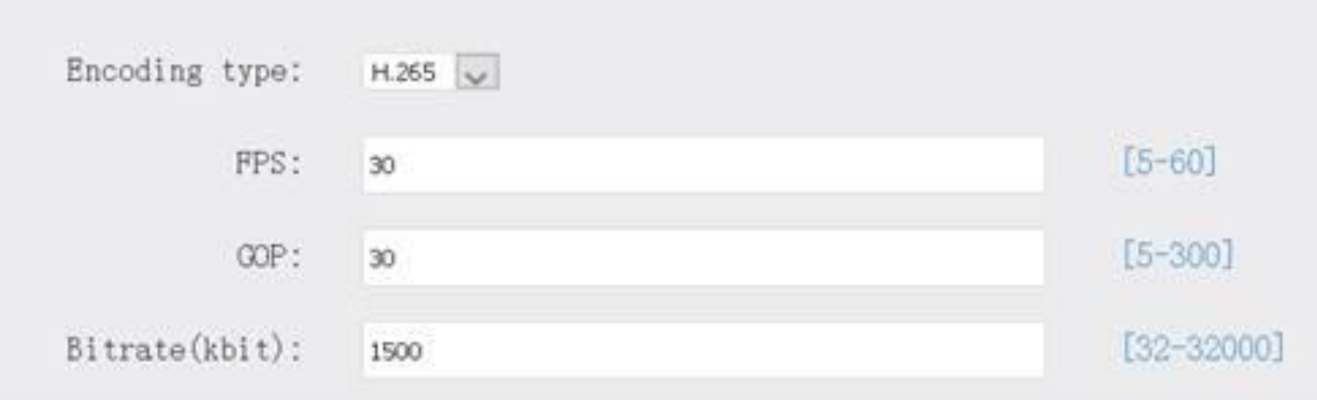


Now with backup and upload function for the firmware itself and the Encoder settings.
Please note:
The ***box.ini*** is a text-file which is Linux conform- (i.e., CR/LF Carriage Return and line feed are different). So, if like to edit it and upload back **do not use a windows-based** text editor but notepad++ (freeware to download for WINDOWS) is working:



Also, to notice: here are all configurations inside which allows changing anything even to values for the encodings which might be out of range if you change those values.

So, keep them in the operational mode ranges like the text in the Web-If is describing like:

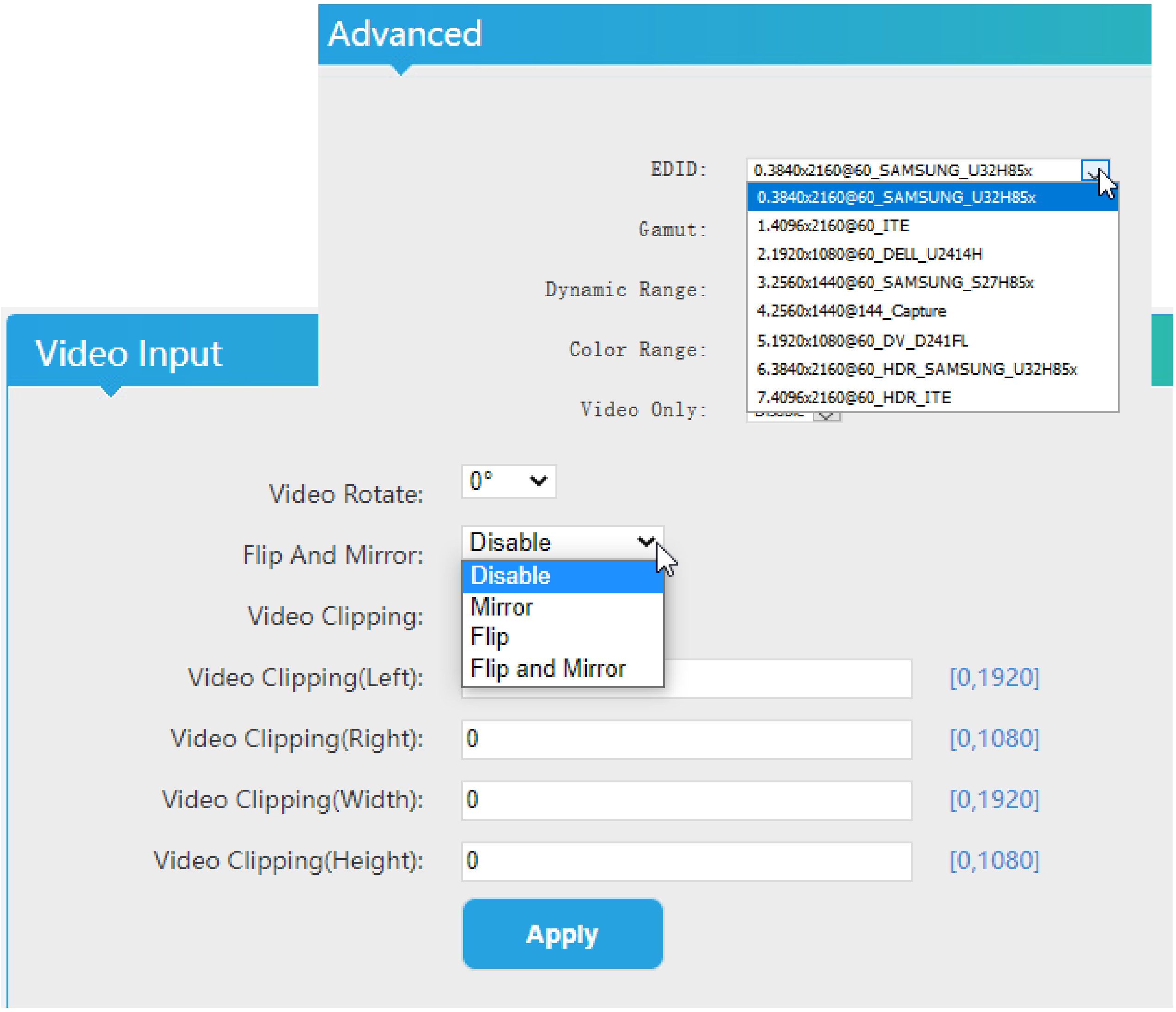


Do not modify any value where you are not 100% knowing what you are doing.
Because the firmware is specific for every model, HDMI or SDI with or w/o h.264 h.265 do not mix up them because the filename of the firmware is always up.rar .
Please use the backup only for saving a firmware for accidently cases where it might be necessary to go back a version.
Danger: If manipulating the up.rar firmware compressed file, do not use a modern 64bit rar packer because that will not work. 7Zip 32/64 will also not work.
So, we recommend not to pack that wrong because it can brick the unit and makes it non accessible any more.
Please use the backup only for saving a firmware for accidently cases where it might be necessary to go back a version. Like backing up before updating a unit…
If you accidently shoot it up, try the RST-RESET Button (5-10 sec. pressing until LED = off) to reload factory defaults.

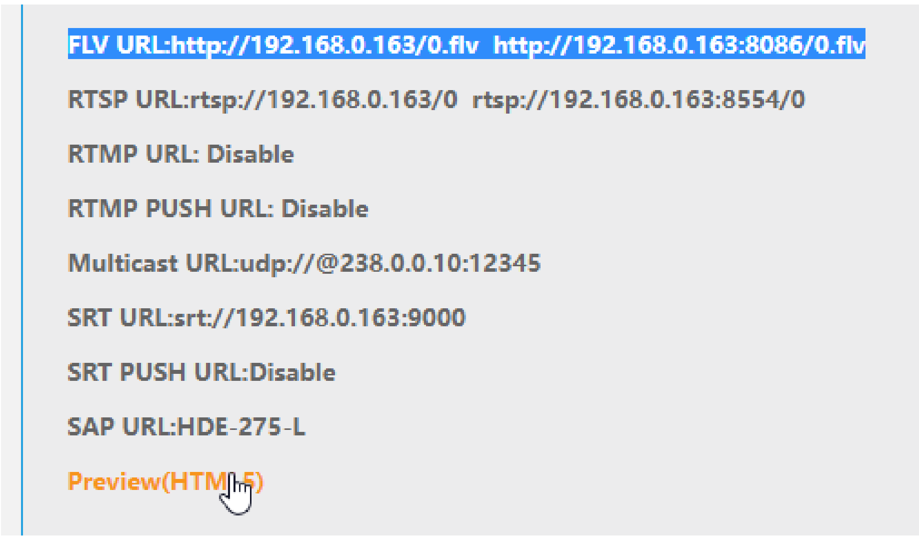
Note:

Some encoder types allow to change the output Angle/direction of the picture by 0° -> 90° -> 180° -> 270° which is useful for Signage Displays…

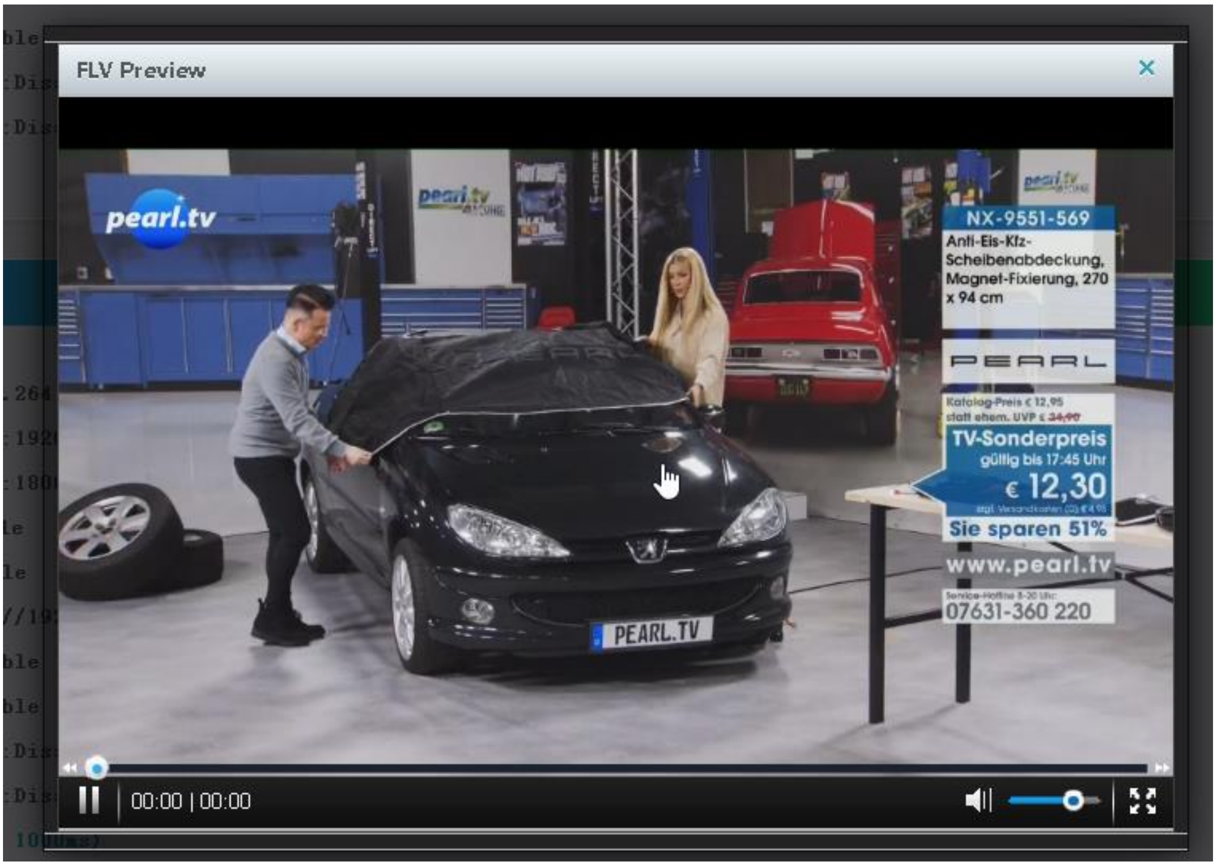
The 4K encoder also have EDID settings which can be changed and adjusted according to the 'TV set'.



Depending on Model: **Preview in Browser** is possible from within the status page as a link:

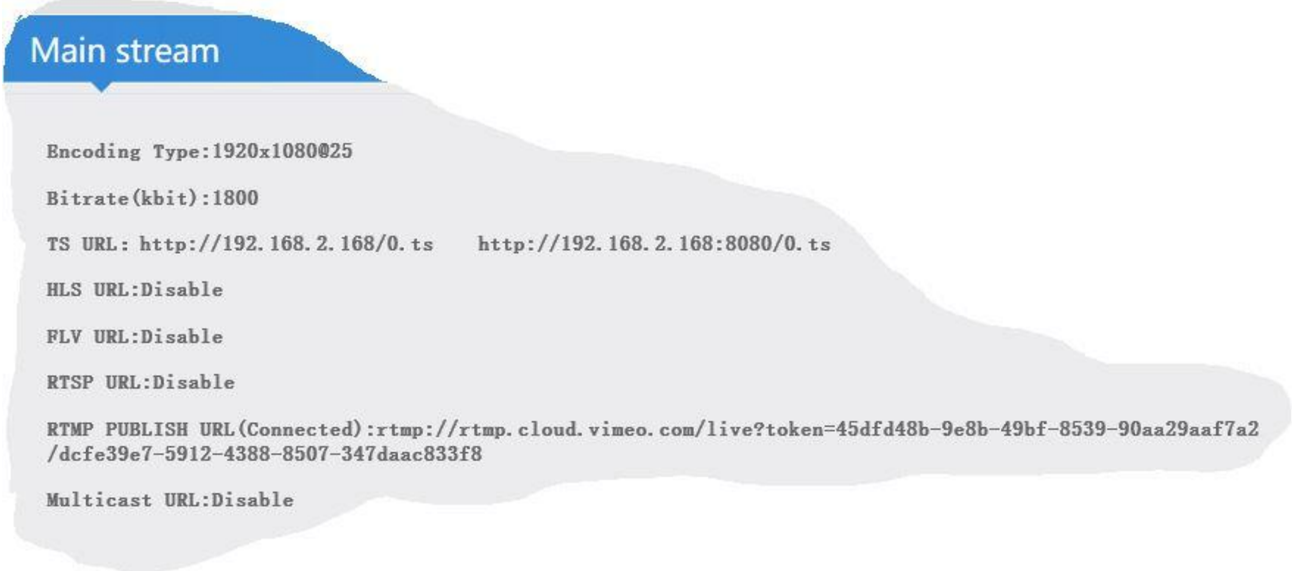


Your Codec should be set to h.264 because h.265 is not guaranteed to work with the Preview player.



But anyway, Flash is now history. But note: Some protocols are not capable of carrying particular Codecs…

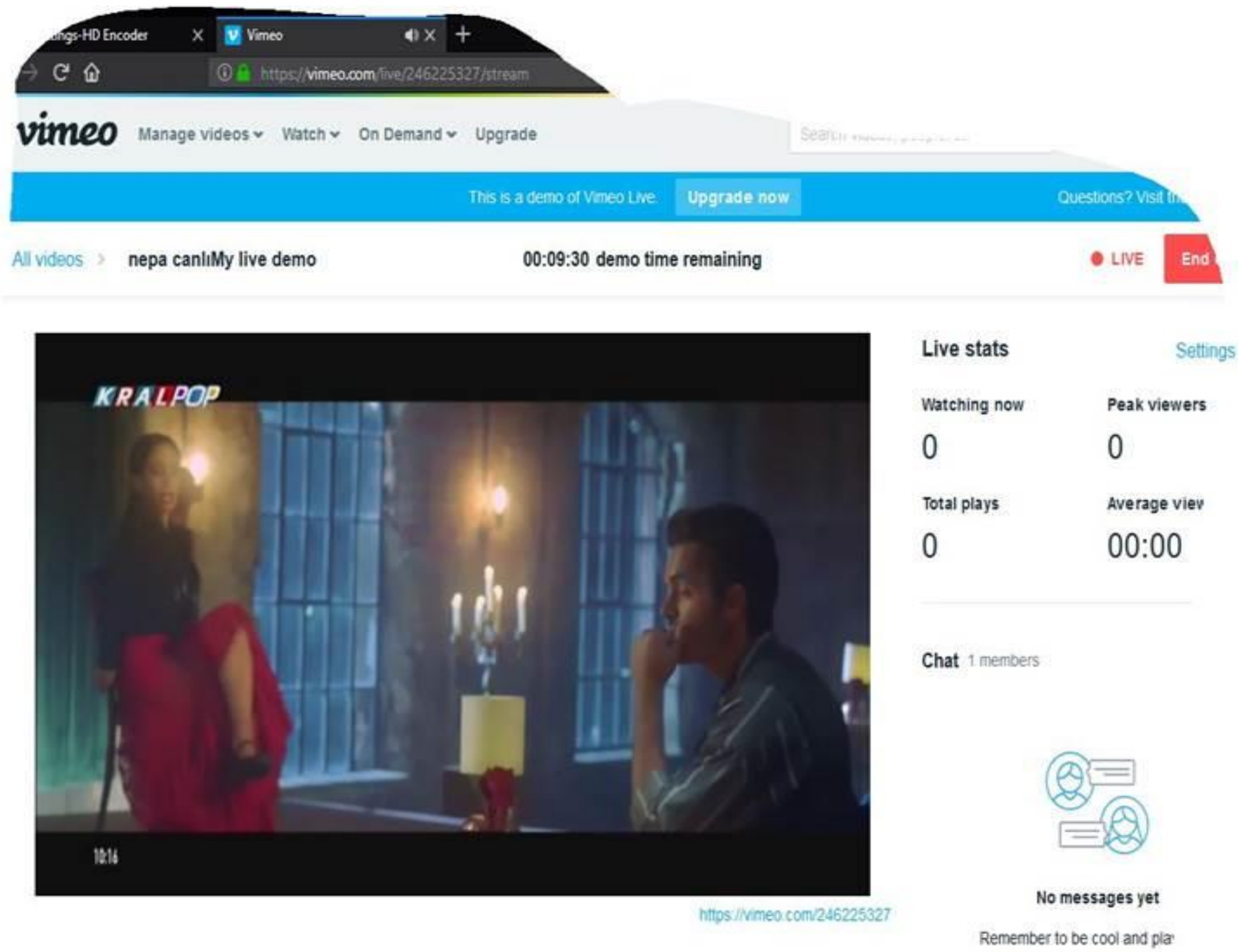
Example for streaming to VIMEO Live by RTMP:



VIMEO gives the user an RTMP –address with a live token at the end. No username/password is necessary because they handover individual stream-keys which simply needs to be inserted as

rtmp://rtmp.cloud.vimeo.com/live?token=***********/streamkey**

Then you can control it by checking the **Vimeo live portal** of your stream:

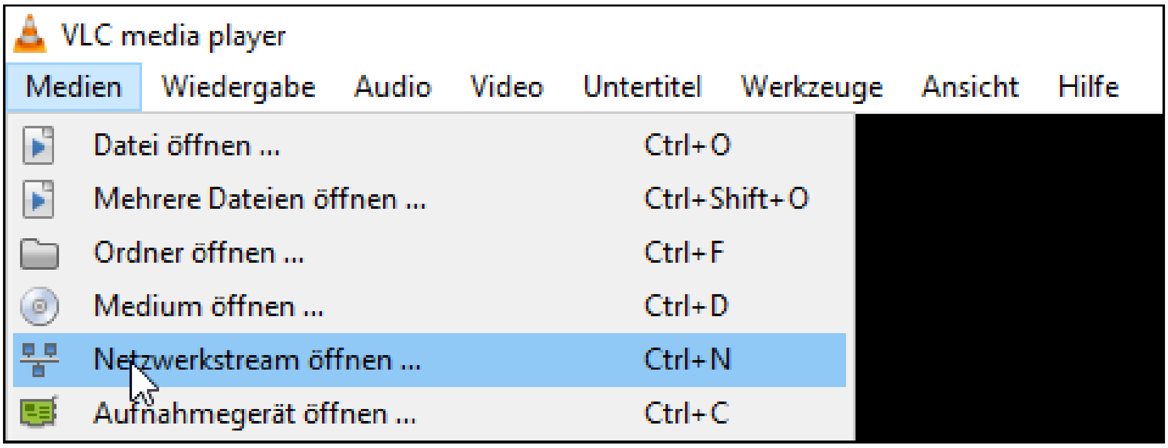


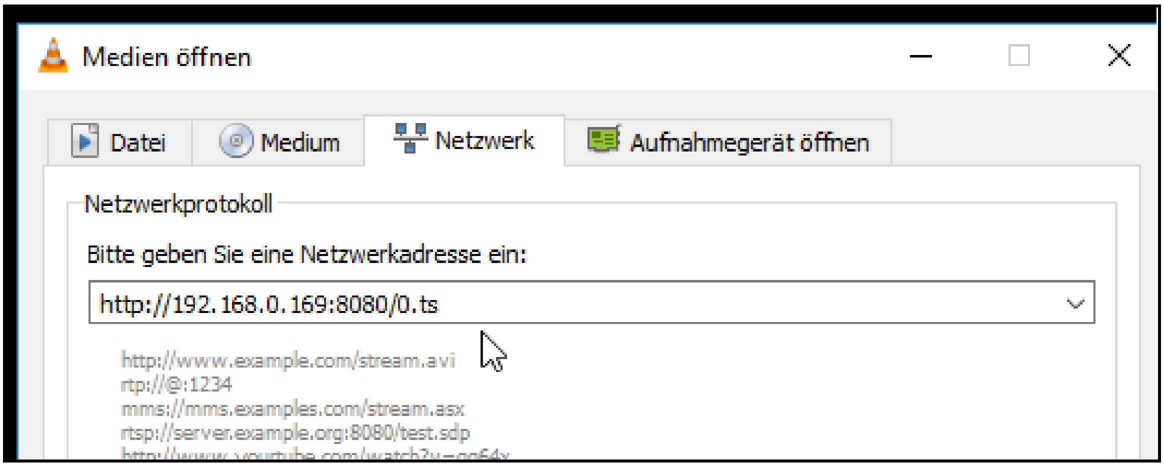
Main stream Live View:

You can play the stream address by your computer if you installed i.e., the VLC Player software or use an IPTV STB by setup the RTSP or HTTP stream addresses:

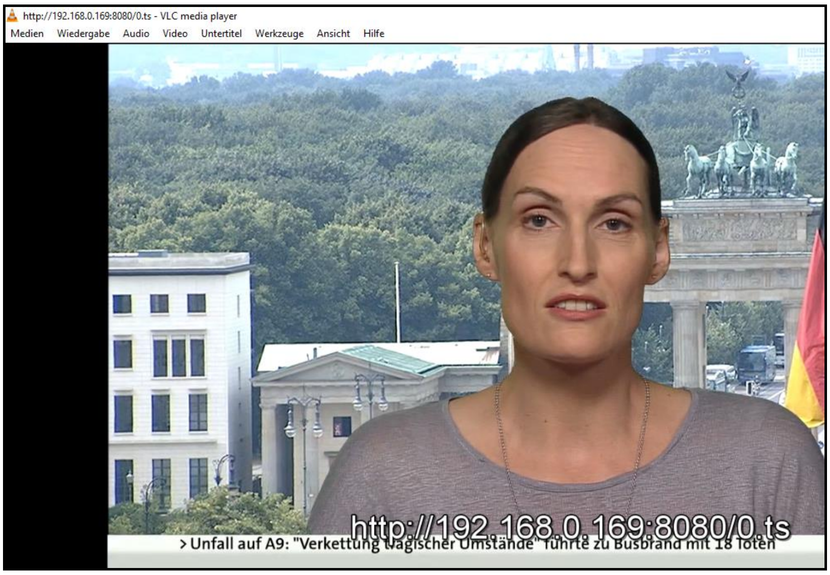
Easiest way: copy and paste the URL from the main window

and open VLC – network stream:

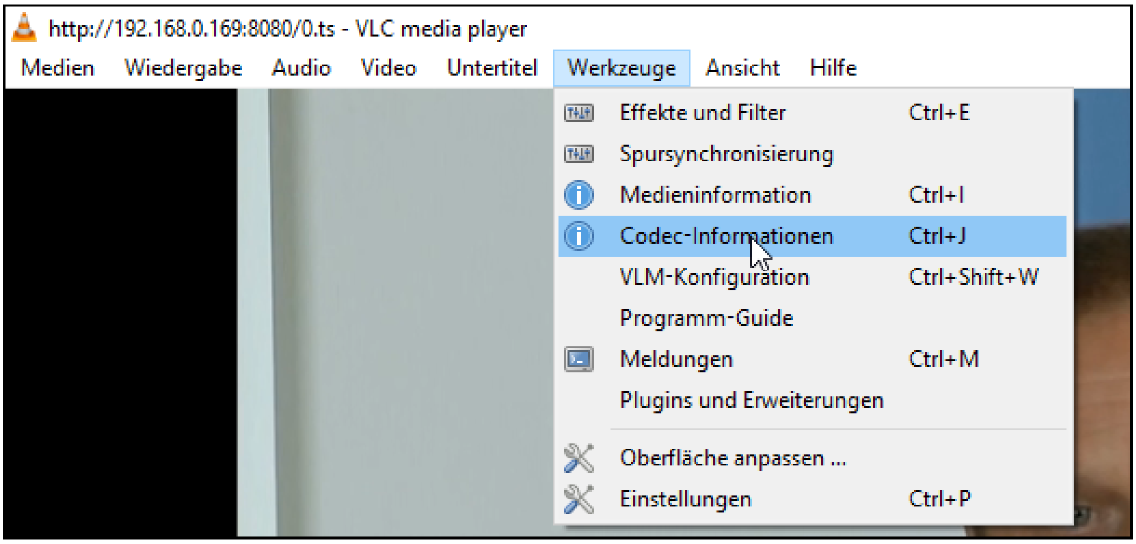




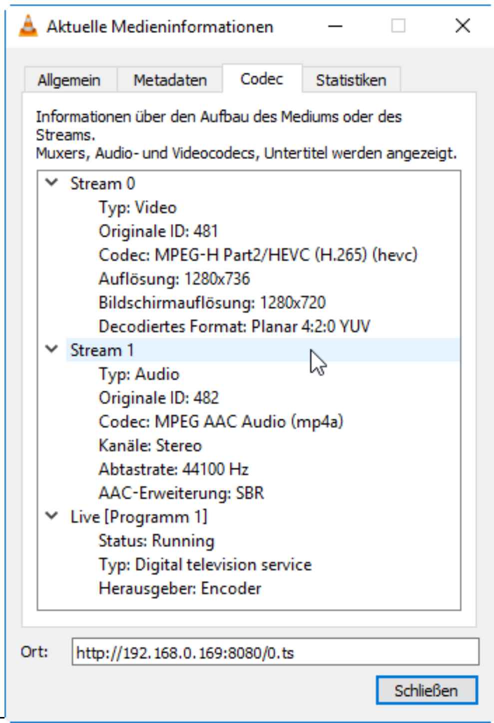
Press Play/Wiedergabe:



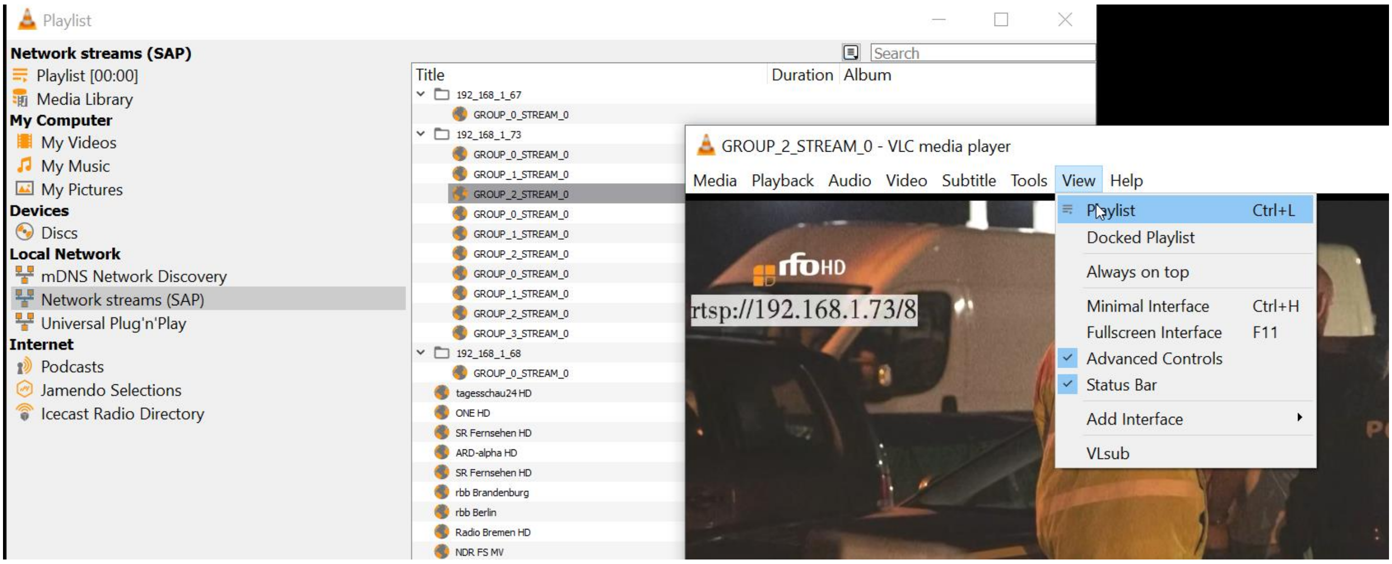
You can check the video information if you like by VLC:



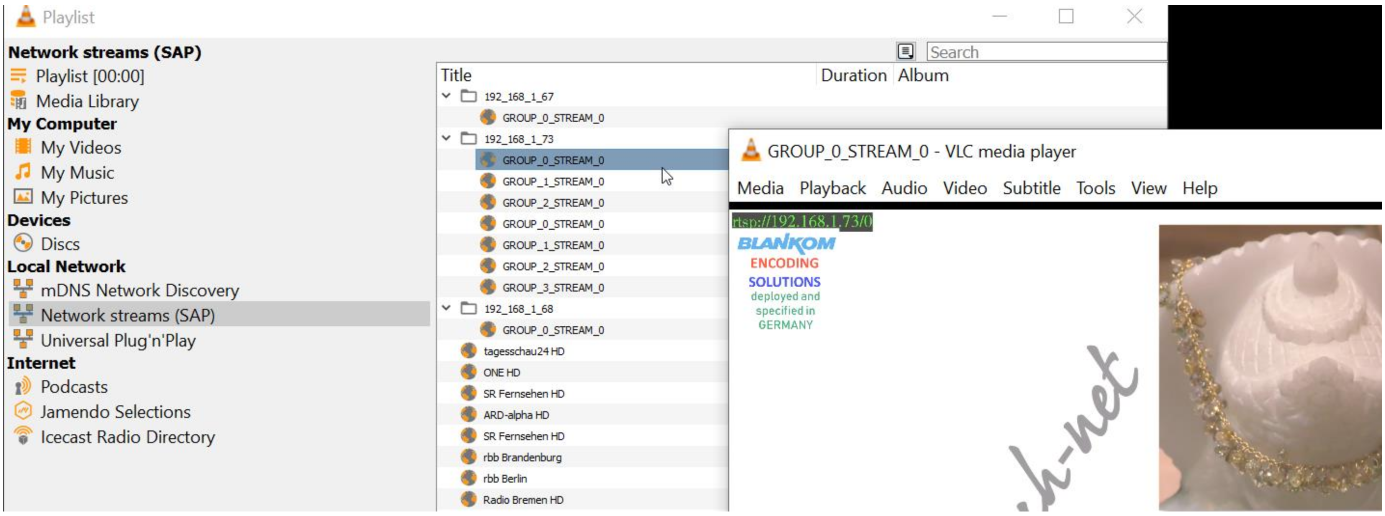
Note: VLC can only receive the stream if your receiving device has only one Network Card enabled! If you have a Laptop with WIFI and Ethernet enabled it doesn't know where to catch the stream from. You can change that by disabling one device or adjust a priority by setup different METRC Values to each of them.



Another Tip: If open VLC Playlist you can use the SAP gathering to receive a network info:



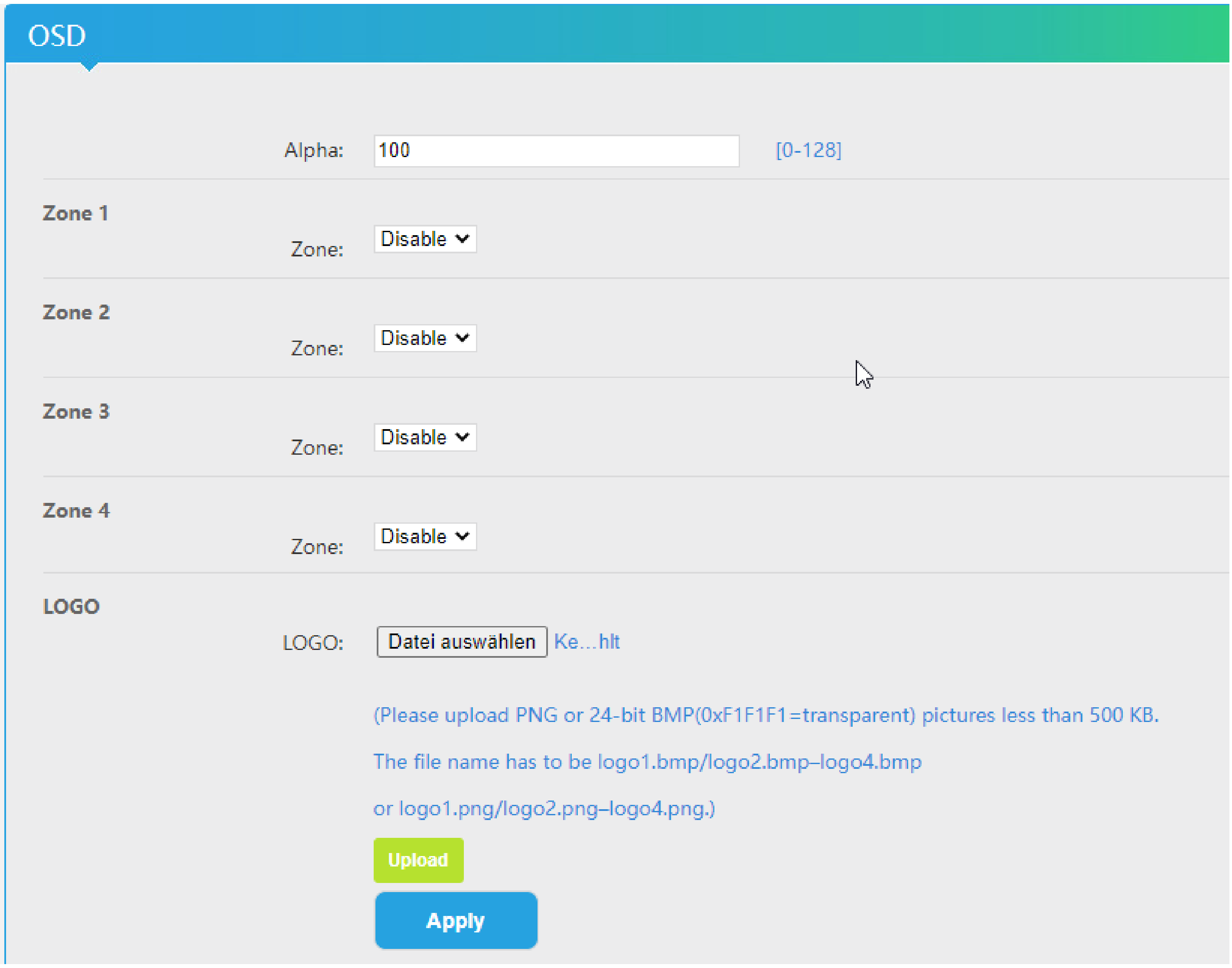
Just double click on that and VLC opens the stream:



SAP is a Session Announcement Protocol that announce on a particular multicast address the streaming info's…

OSD Settings (Overlay a Picture/TXT to the encoded Stream)

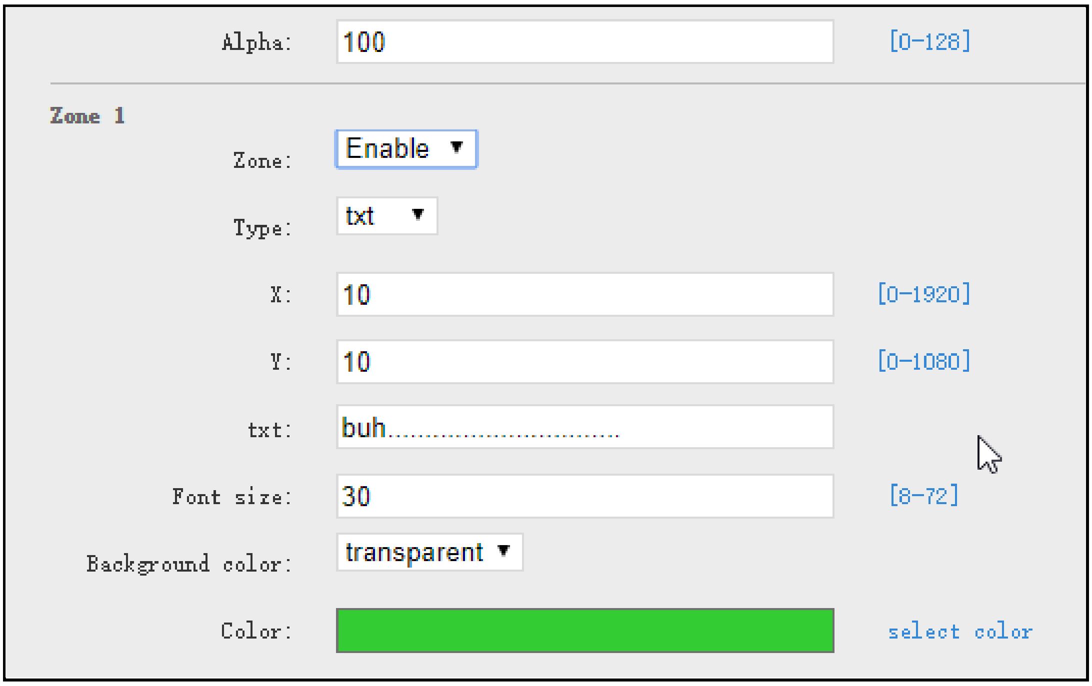
To be able to OVERLAY text and advertisements to your encoded stream, the unit supports up to 4 zones:



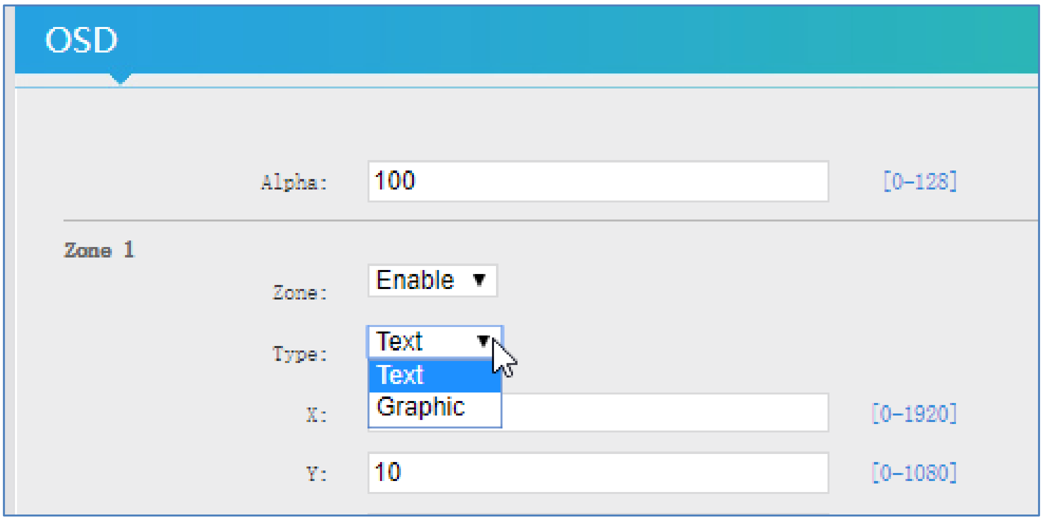
Note: You can insert two couples of TEXT and 3 pictures simultaneously overlaying the picture

Text X:	0-1920 is optional, display the left and right position of the text
Text Y:	0-1080 is optional, display the up and down position of the text
Font1 size:	8-72 is optional
Alpha1:	0-128 is optional, Alpha-blending – transparency setting
Color1:	choose the colour you want to display
Bg1:	choose background colour for the text on the video overlay
Text:	input the content of the text you want to display

These features are varying depending on model and versions.



Or a well-prepared picture according to the values given in the WEB-IF for uploading to the unit:



Since 2019 software version, the Overlay-Logo insertion can be used as PNG transparent or BMP-pictures.

OSD insertion Picture Setting

Picture1…4:	disable/ enable (disable: no images, enable: insert the images)
Pictures X:	4-1920 is optional **to set the** left and right position of the picture
Pictures Y:	4-1080 is optional to set the up and down position of the picture
Alpha:	0-128 is optional - transparency setting
Picture name:	display the name of the picture1
Upload picture:	choose to upload the image, supporting *.bmp format of the picture and limited file size: less than 500kbyte

Requirements

- The settings of the three pictures to be inserted must be identical.
- Transparent background of the picture setting:
  should be of RGB values: R - 177, G - 204, B - 233 or see WEB-IF hints

Example:

(Please upload PNG or 24-bit BMP (0xF1F1F1 is the transparent colour taken) pictures less than 500 kByte, The file name is logo1.bmp or logo1.png and so on according to the inserted 4 zones logo2...4):

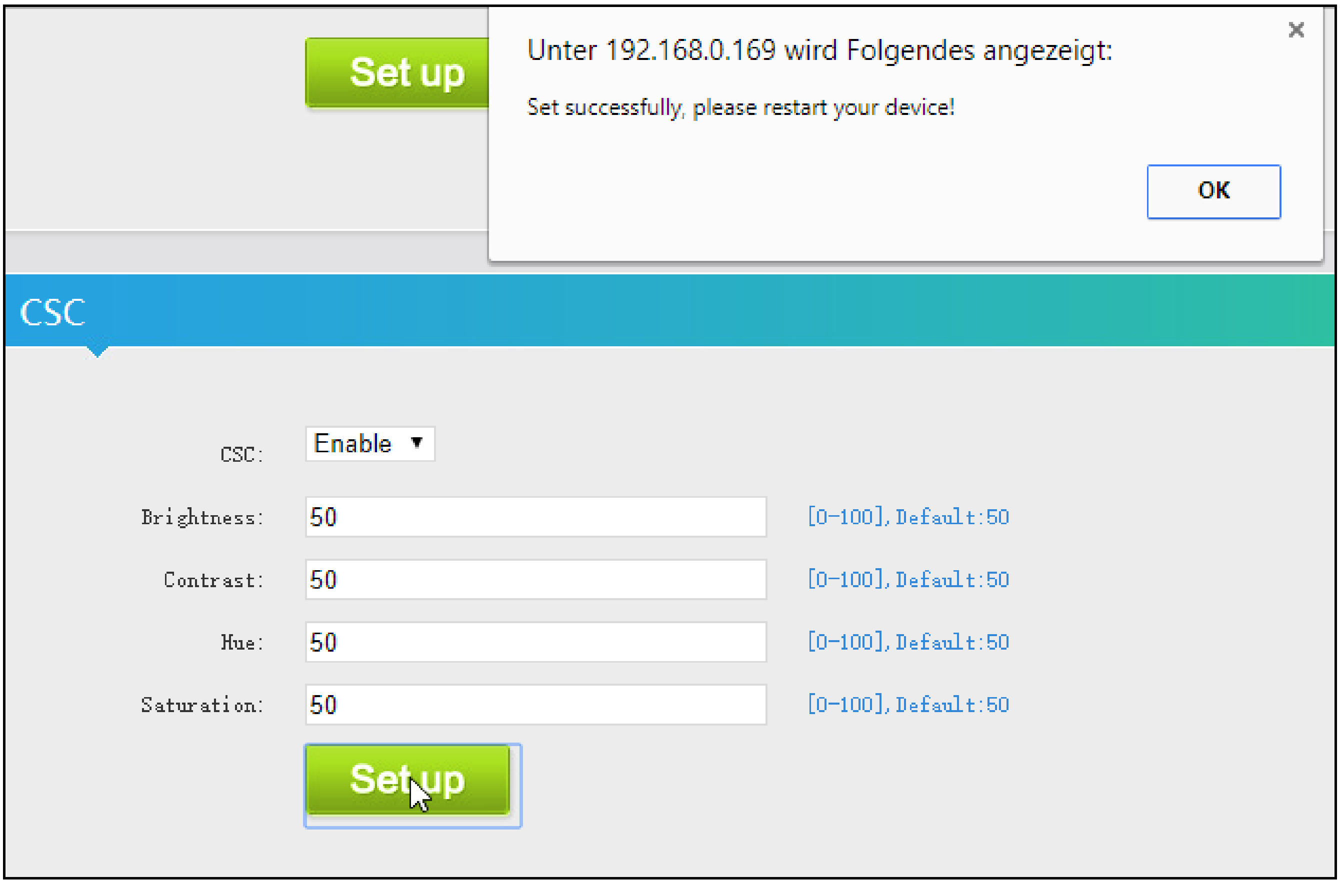
Example: The bitmap BMP:

Komprimierung:	Keine
Auflösung:	300 x 300 DPI Ändern
Originalgröße:	432 x 89 Pixel (4.85)
Aktuelle Größe:	432 x 89 Pixel (4.85)
Druck-Größe (aus DPI):	3.7 x 0.8 cm; 1.44 x 0.30 inches
Originalfarben:	16,7 Millionen (24 BitsPerPixel)
Aktuelle Farben:	16,7 Millionen (24 BitsPerPixel)
Gezählte Farben:	246 ☑ Zählen aktiv
Benötigter Plattenplatz:	112.69 KB (115.398 Bytes)
Benötig. RAM-Speicher:	112.68 KB (115.384 Bytes)

The light grey background colour is: 0xf1f1f1 and will appear in the TV screen as Transparent.

You can use GIMP or any other graphic software to change your logos background accordingly. PNG has a transparency option – BMP doesn't.

HINT: If you change parameters or enable features, following popup message will appear:



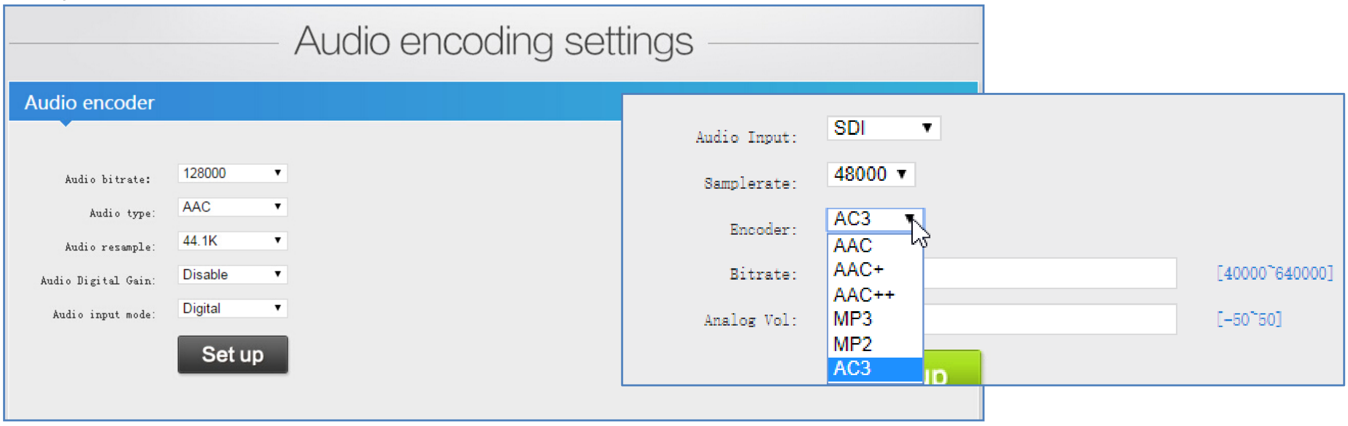
If this above popup message appears:

This doesn't mean to restart your encoder! – It means your **Receiver** should be tuned again (restarted) to the stream i.e., switch or reload the channel on your IRENIS/BLANKOM IPTV SetTopBox to re-initialize the decoding process or restart VLC.

Audio Encoding Settings

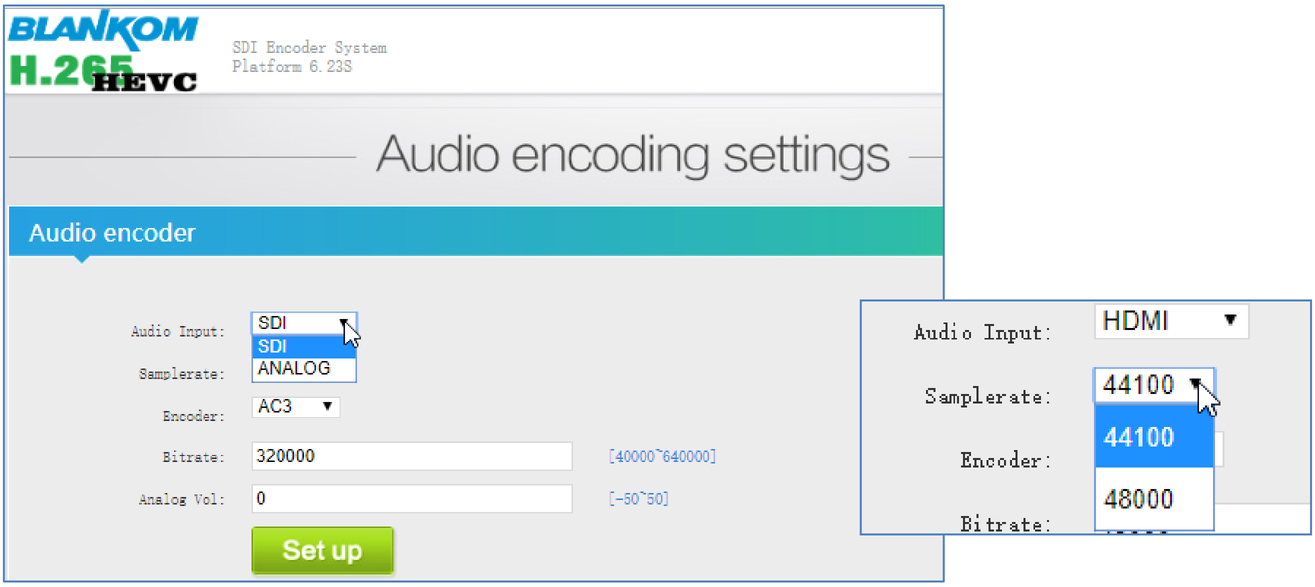
Depends on Model
Audio bitrates: 48k, 64k, 128k, 160k, 192k, 256k (depending **on chosen following** codec)
Audio type: AAC ... (depending on model) and more...
Audio digital gain: 2x, 4x, 8x, used to adjust volume
Audio input mode: digital/analogue

Example 1:



Audio codec support depends on Model and SW-Version

Example 2:



SOME MODELS ALSO HAS AUDIO-SYNC SETTINGS …

TECHNICAL SPECIFICATIONS (dep. on Model – see separate data sheets)

Video

Input	HDMI, SDI, CVBS according to the chosen model
Resolution	1920×1080_60i/60P, 1920×1080_50i, 1280×720_60p, 1280×720_50p and below
Encoding	h.264/AVC Main Profile/High Profile; H.265/HEVC Baseline Profile;
Data Rate	0.8 Mbps ... 12 Mbps (32kbs ...32Mbps)
Rate Control	CBR/VBR
GOP Structure	IBBP
Advanced Pre-treatment	De-interlacing, Noise Reduction, Sharpening

Audio

Encoding	AAC (+, ++...), MP3, AC3, MP2/3 ... dep. on model
Sampling Rate	Auto
Bit-rate	48K/64K/96K/128K/160K/192K/256k
Sampling Precision	16 bit
Data Rate	64 Kbps ... 384 Kbps

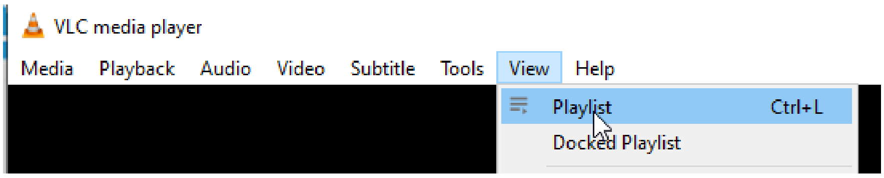
System

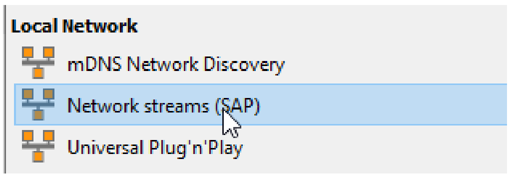
Operating System	HiLinux embedded OS
Ethernet/RJ45	100BaseT (h.264 only version), 1000Base-T RJ45 h.265 versions
Protocol	HTTP, UDP, RTP, HLS, RTSP, RTMP, ONVIF (prot.: S,C,G)
Control Interface	100/1000BaseT by WEB-Browser

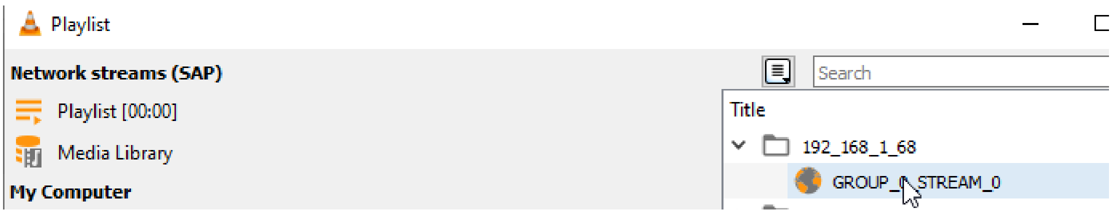
We recommend to make yourself familiar with 'What is Multicast and Unicast' and the corresponding IP-Ranges.

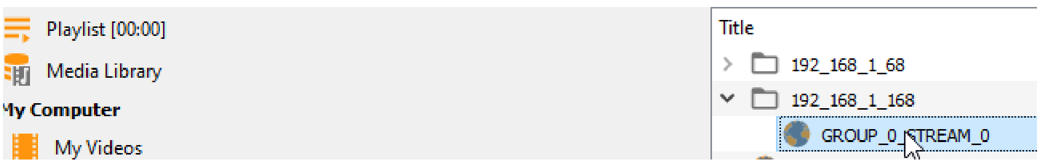
SAP-support for Multicast-streaming:

Using VLC SAP-Gathering will show a simple click'n start entry:





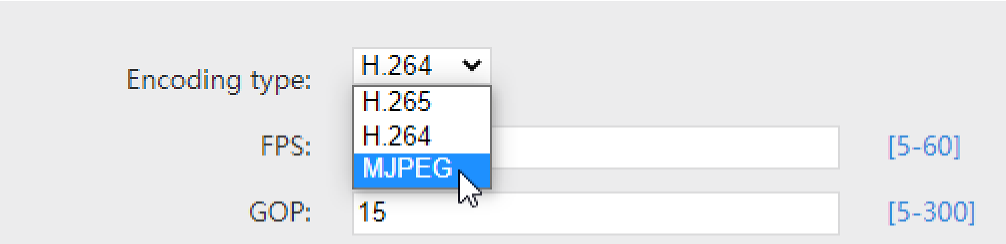




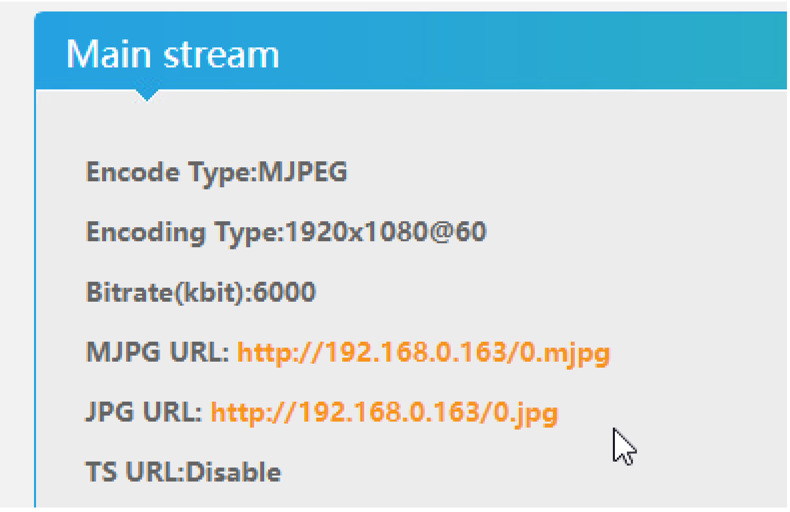
-> Will receive the stream. This works only with Multicast UDP / RTP!

A seldom case but: **MJPG support:**

If you directly want to send the 'pictures' only as motion JPG format to a browser, you can set this to be enabled:

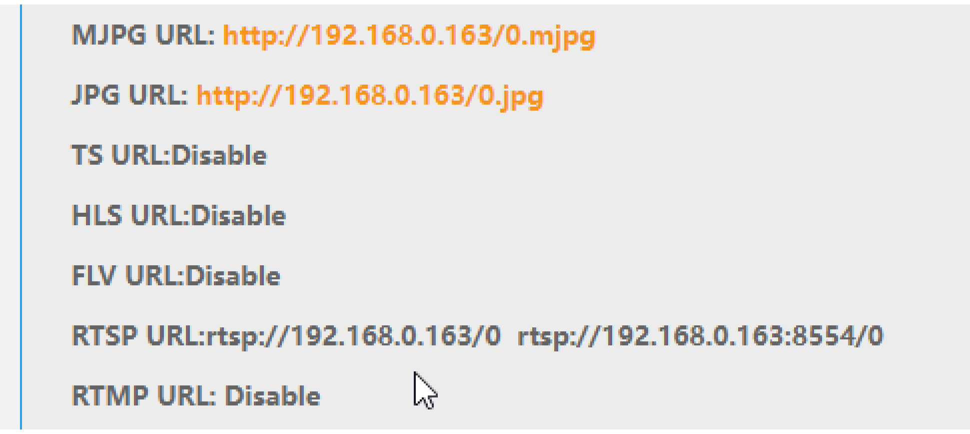


We recommend better to choose the **Main-encoder** part for this so the status page will show:



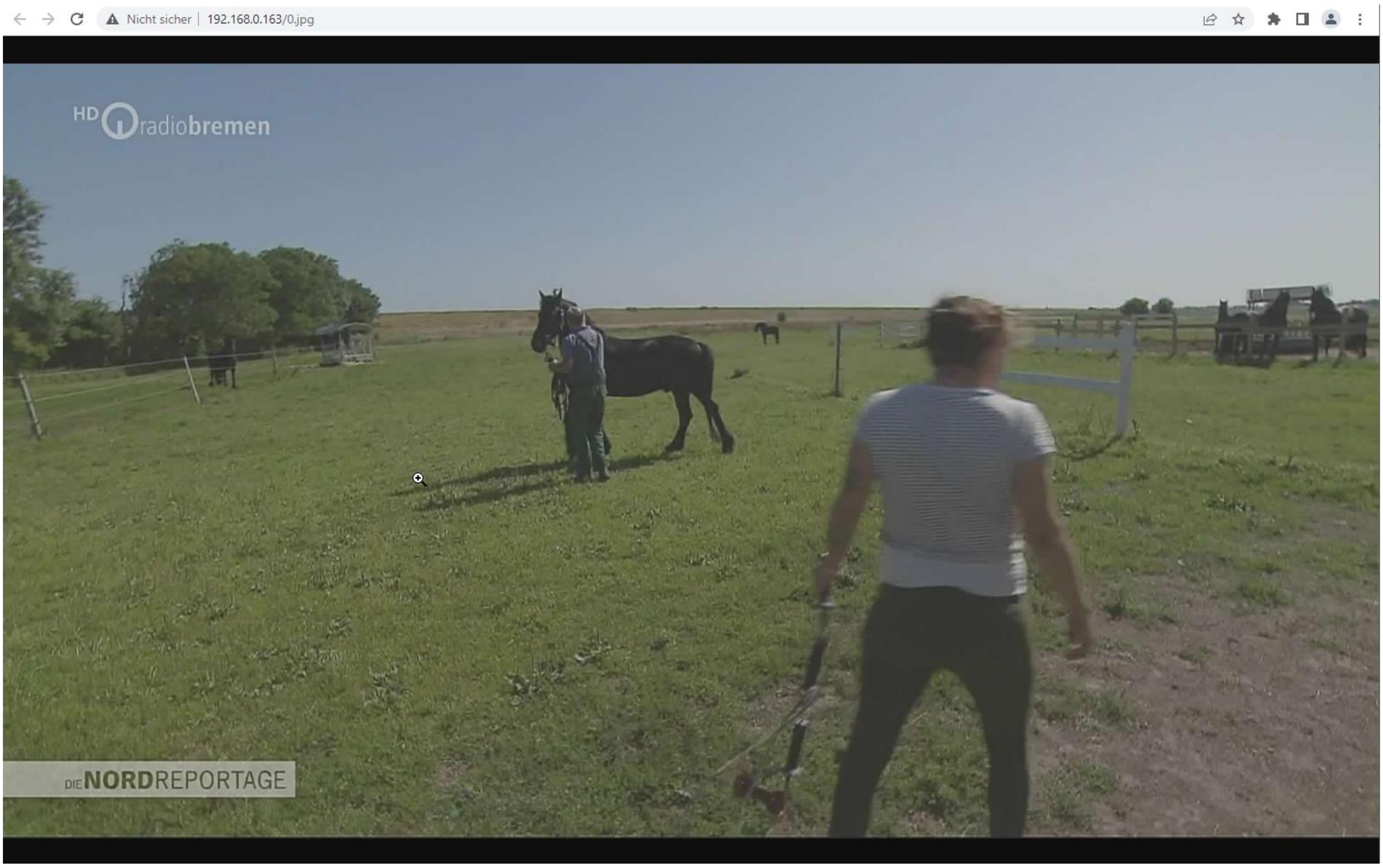
Please enable at least one RTSP output before changing to MJPEG – otherwise no streaming will happen.

-> Status page... **PLEASE Note: RTSP has to be enabled for MJPG-stream:**



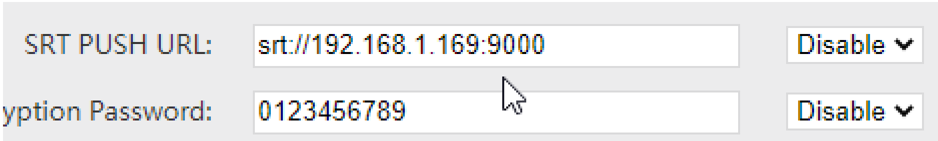
Link open by click and your browser opens it:

Or only the still picture shows the moment of the screen when clicking on /0.jpg:



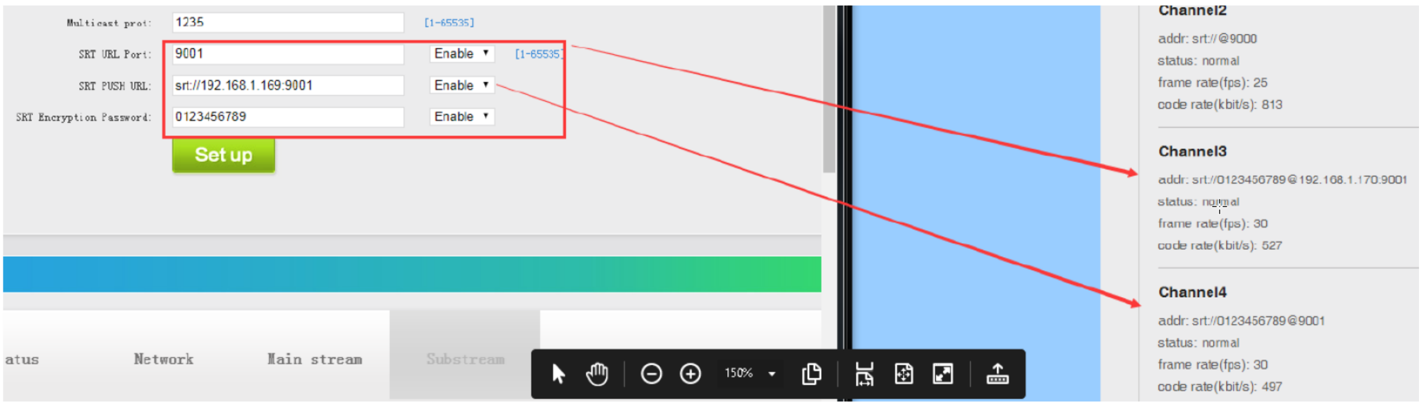
SRT-Support:
(Only supported by our encoders with h.265 compatibility because of processing power)
What is an SRT? Secure Reliable Transport (SRT) is an Open-source software protocol and technology stack designed for live video streaming over the public internet. SRT provides connection and control, reliable transmission similar to TCP, however, it does so at the application layer, using UDP protocol as an underlying transport layer. It supports packet recovery while maintaining low latency (default: 120 ms). SRT also supports encryption using AES. Source: https://en.wikipedia.org/wiki/Secure_Reliable_Transport Note: SRT works only in pairs: The stream receiver must support SRT reception. Video Encoders are widely used in video transmission field, and SRT supported by our video encoder & decoder. Our Encoder & Decoder work perfectly for Haivision Play, Larix Broadcaster, etc. More details: https://www.srtalliance.org

SRT-live-server (SLS)-for our Video Encoder
Our Video Encoders support SLS for SRT.
Introduction
srt-live-server(SLS) is an open source live streaming server for low latency based on Secure Reliable Tranport(SRT). Normally, the latency of transport by SLS is less than 1 second via the internet.
Requirements
Please install the SRT first, refer to SRT(https://github.com/Haivision/srt) for system environment basics. SLS can only run on OS based on linux, such as mac, centos or ubuntu etc.
Source: https://github.com/Edward-Wu/srt-live-server
Put the following url to send to your docker container: srt://your.server.ip:1935?streamid=input/live/yourstreamname

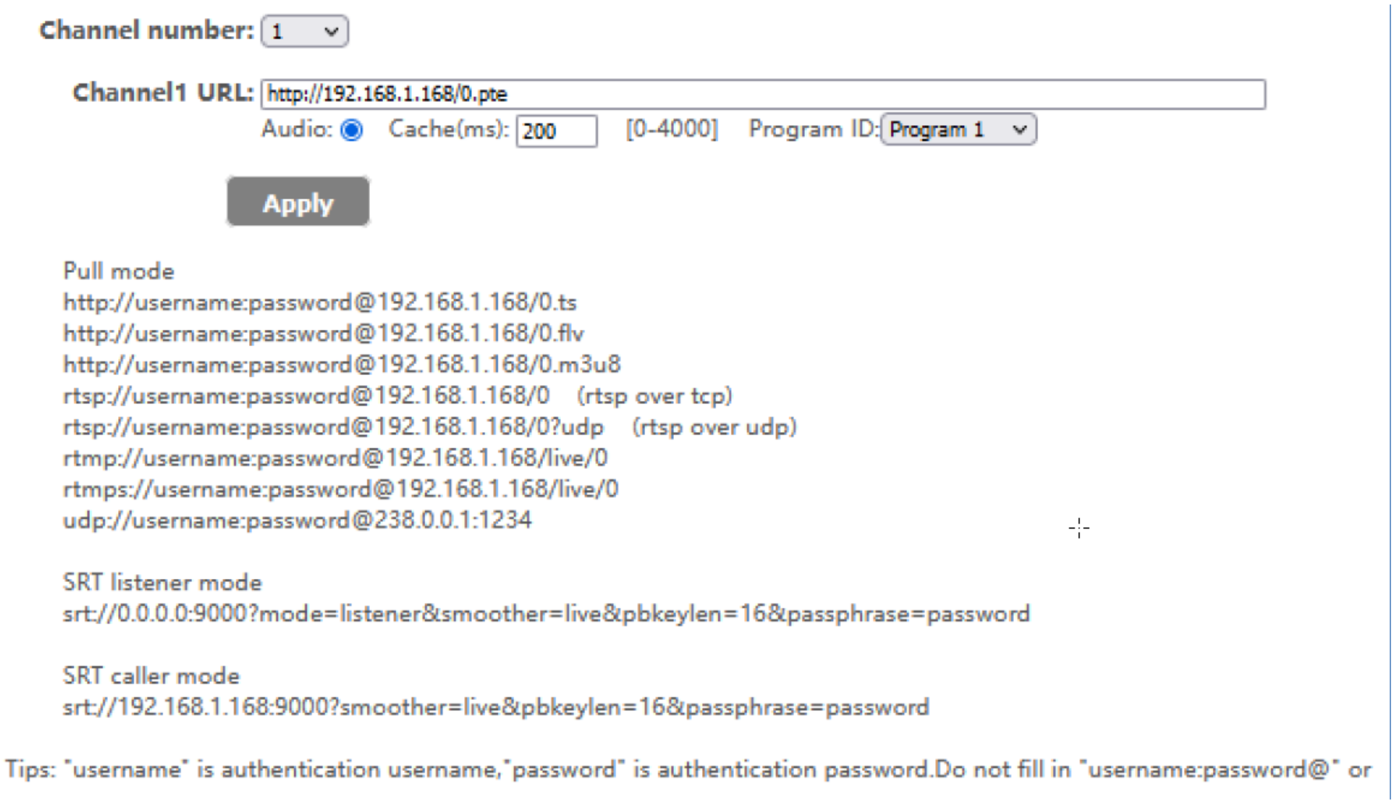


Video Encoder & Decoder SRT settings as couple:

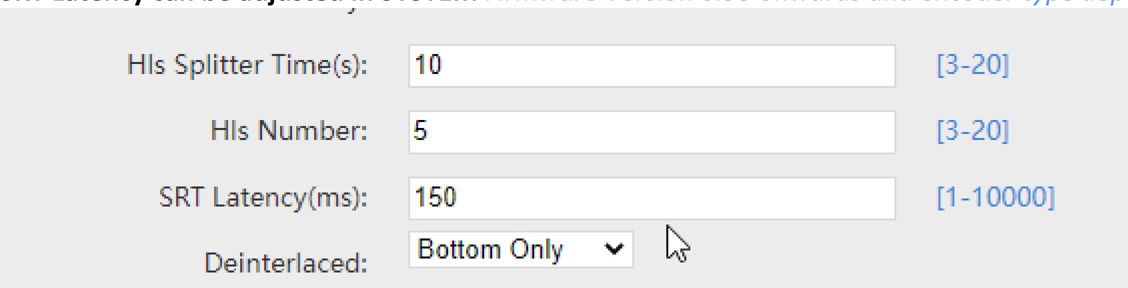
For HDMI/VGA&CVBS/SDI Decoder-Support h264 & h265, decoder SRT playing the URl as, here the encoder works as caller (SRT push URl) and listener (SRT URl port):
srt://ip:port ***# encoder as Listener, decoder get srt from encoder, here 'ip' is the Encoder IP.*** srt://port or srt://@port ***# encoder mode as caller, push SRT to the decoder, (encoder SRT push URl as srt://decoder ip:port)*** With passphrase/Encryption, decoder SRT play URl:
srt://passpharese@ip:port ***# encoder as Listener, decoder get SRT stream from encoder, here 'IP' is the Encoder IP.***
srt://passphrase@port ***# encoder mode as caller, push srt to the decoder.*** See below screenshot for settings:



It is a little complicated … Setting hints in the Decoder Web:
Like the user-password encoded streams in
Pull mode http://username:password@192.168.1.168/0.ts http://username:password@192.168.1.168/0.flv http://username:password@192.168.1.168/0.m3u8 rtsp://username:password@192.168.1.168/0 (rtsp over tcp) rtsp://username:password@192.168.1.168/0?udp (rtsp over udp) rtmp://username:password@192.168.1.168/live/0 rtmps://username:password@192.168.1.168/live/0 udp://username:password@238.0.0.1:1234
Can be used to receive secured streams from our encoders



SRT Latency can be adjusted in SYSTEM *Firmware Version 6.53 onwards and encoder type dependent…*:



It's a faster transport protocol for lowering latency over (public) networks…

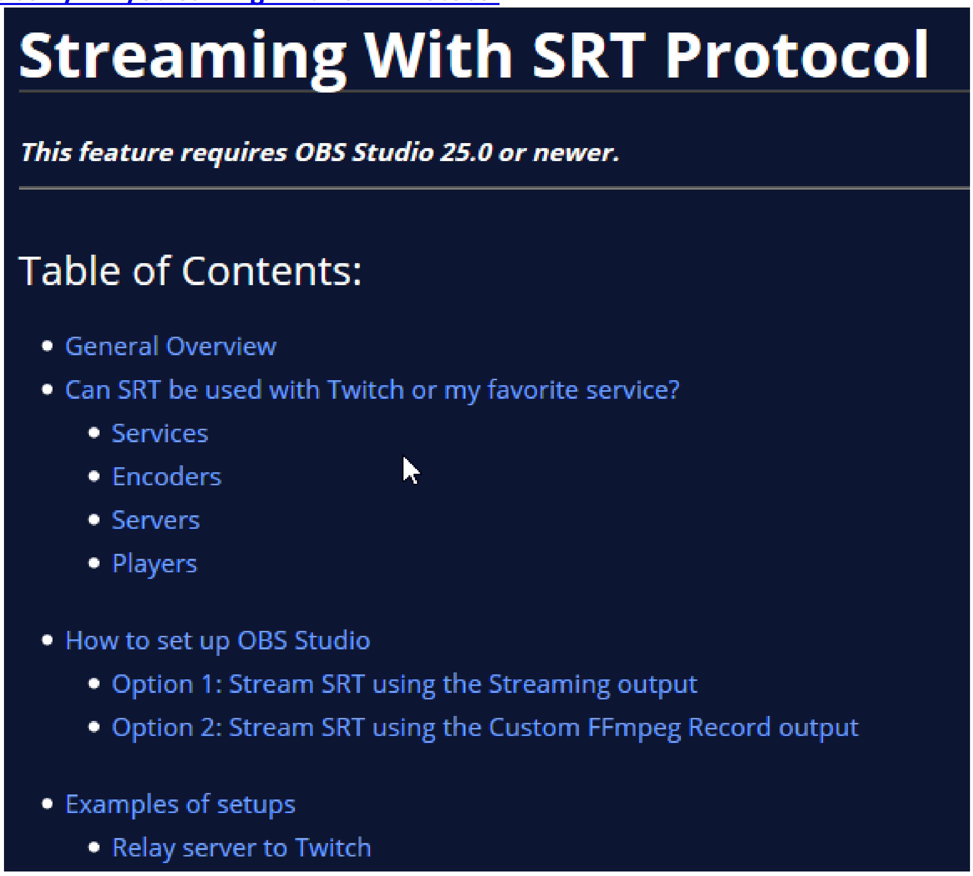
Usually, SRT URL is OK for simple streaming from Encoder to the Client (media player, VLC, STB – but need to have SRT support in the client software).
For P2P direct streaming, select SRT PUSH and enter the destination IP Address and Port. Both source and destination (STB or VLC-PC or Decoder) have to be in the same subnet. Example: Over VPN, both devices need to 'see' each other (i.e., use PING).
You can check it by VLC: *(please note, the @ in the URI is* ***not*** *necessary like in udp/rtp)*

Some more useful links regarding SRT:

A Media server to handle SRT and more: The Open Broadcaster Software

https://obsproject.com/

https://obsproject.com/wiki/Streaming-With-SRT-Protocol:



Streaming With SRT Protocol

This feature requires OBS Studio 25.0 or newer.

Table of Contents:

https://github.com/obsproject/obs-studio

https://github.com/haivision/srt

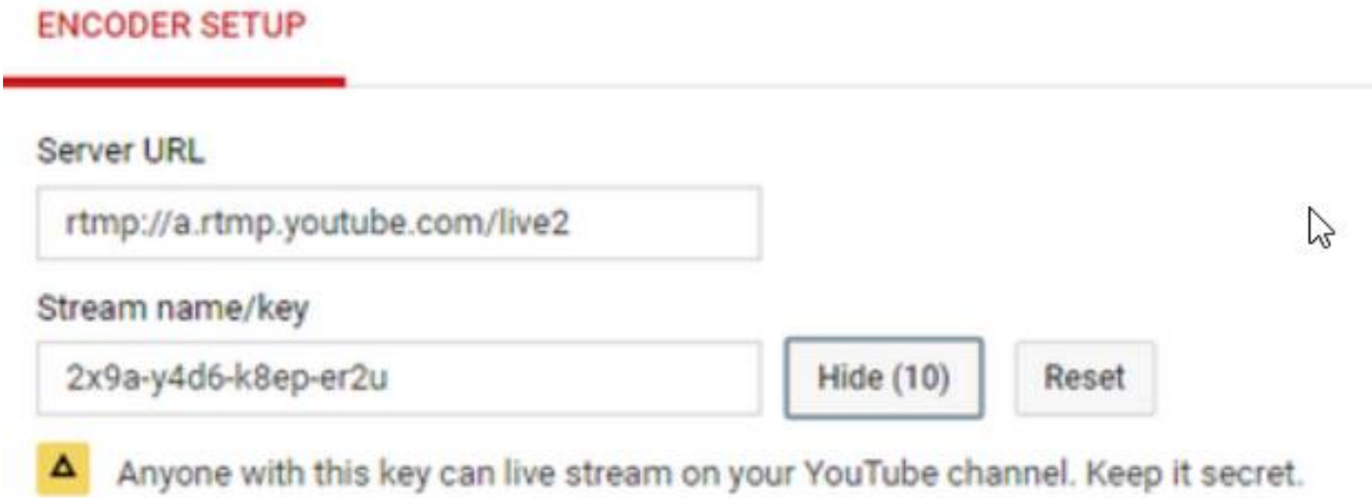
Example to push the encoded stream to YouTube/Facebook

(Depending on Firmware and hardware Versions):

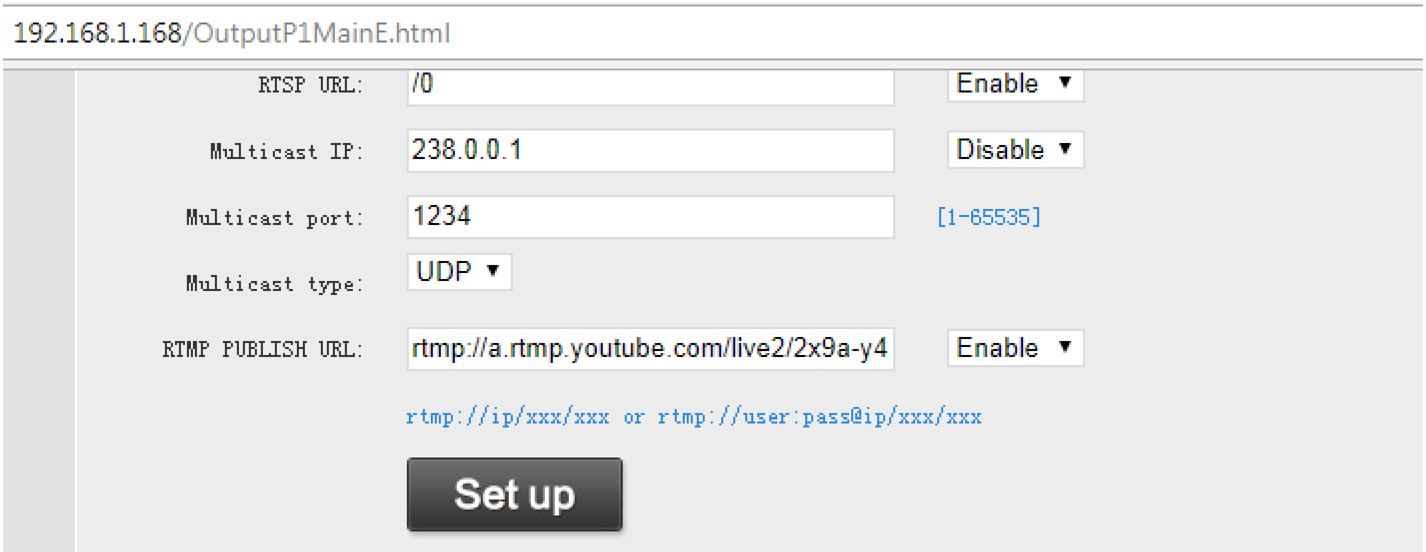
Important, if all your settings are correct, but still doesn't work, please check the encoder network settings, ensure the **DNS settings** are correct because it 'pushes' via your router to the internet as upload to an address which needs to be translated into its IP address:

In Some region, due to local laws and regulations, you may can't send RTMP to YouTube or Facebook.
Our HD/UHD Video Encoder to YouTube Live Stream settings example:
Enter your YouTube account:

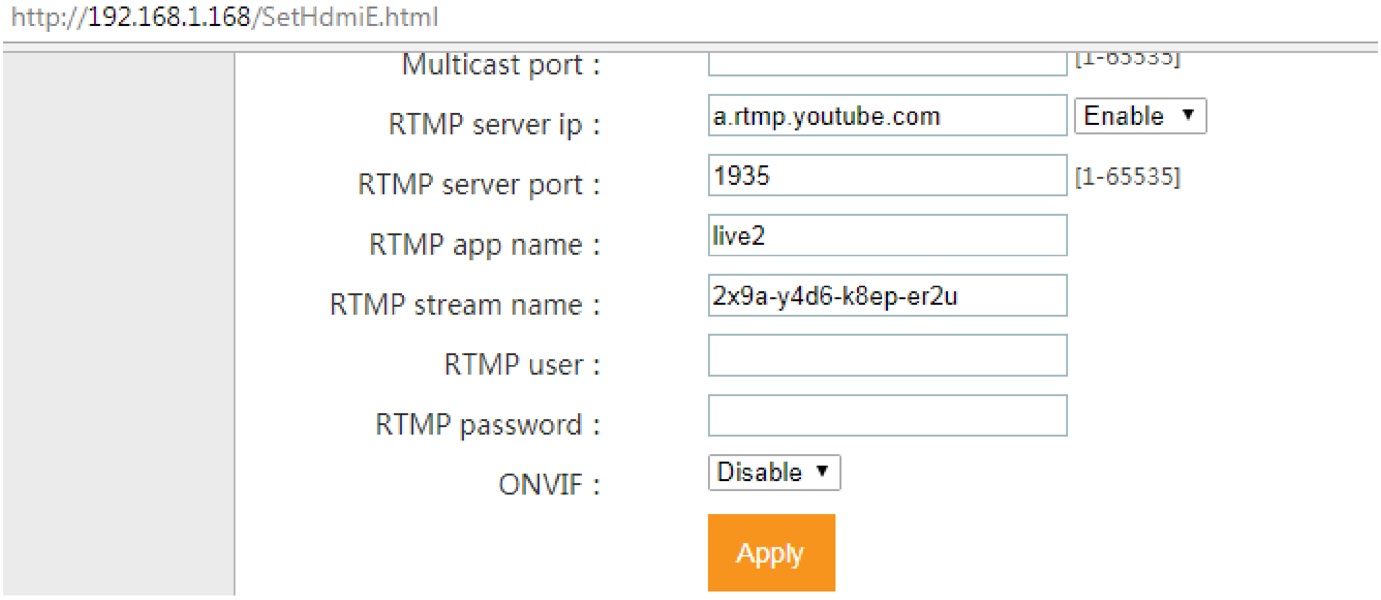
Goto:



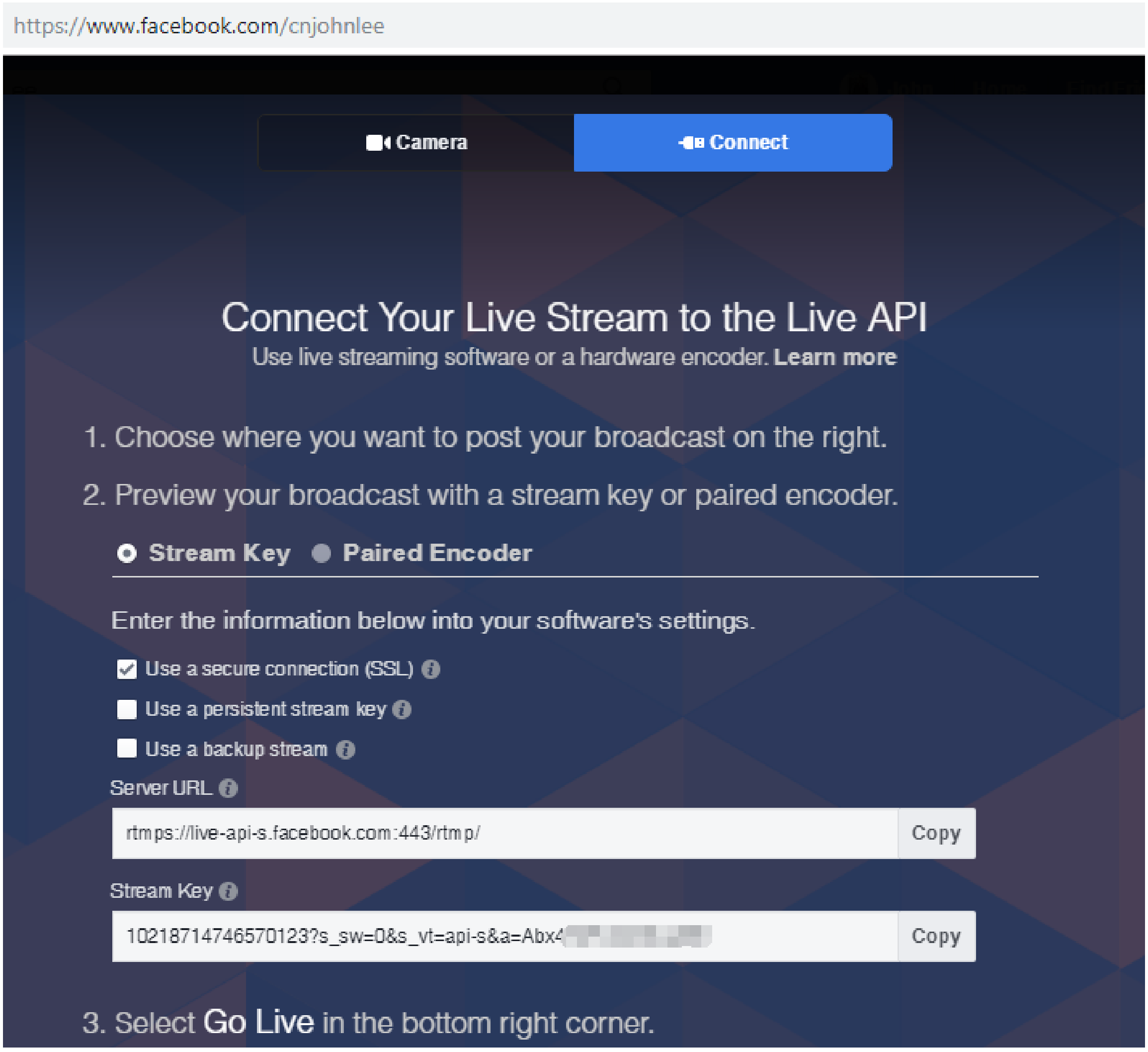
For our newer Video Encoder, such as HDE-275…, etc., the encoder input address for **RTMP** is as example:
rtmp://a.rtmp.youtube.com/live2/2x9a-y4d6-k8ep-er2u *copy and paste this or manually insert it*



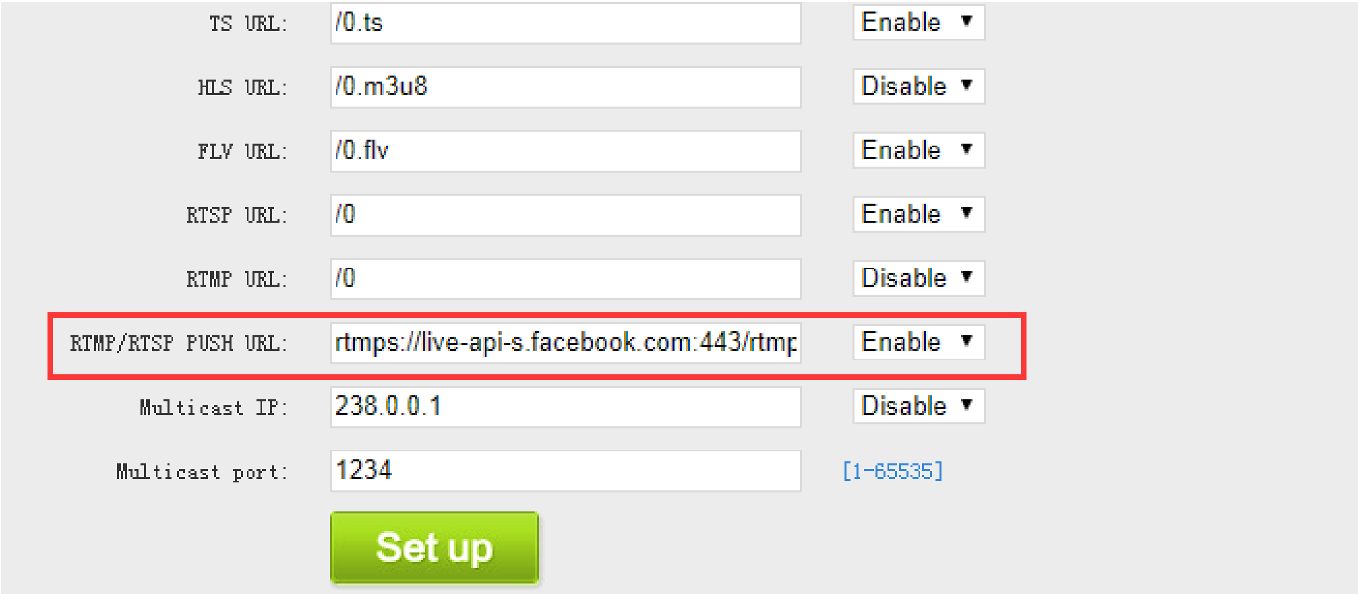
<u>For older Video Encoder versions please use these values</u>
RTMP server ip : a.rtmp.youtube.com
RTMP server port : 1935
RTMP app name : live2
RTMP stream name : 2x9a-y4d6-k8ep-er2u (*which is your individual account key/name)*



New YouTube / Facebook method by RTMPs Video Encoder
RTMPs settings to **Facebook** Live Stream:



For the newer Video Encoder, such as HDE-276, etc., insert this by input the URL
rtmps://live-api.facebook.com:80/rtmp/10214319118682173?s_sw=



Some more Tipps and tricks:

Input- ON-OFF switching's

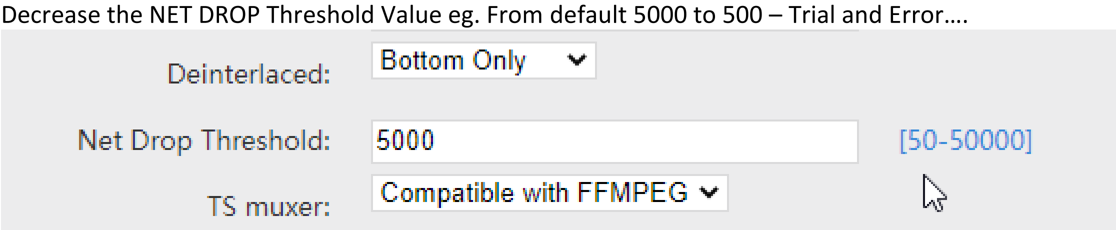
If the encoder is used with switching Inputs like: SDI or HDMI are changing their resolutions and Video parameters during the encoding and streaming processes the Stream will play the 'Signal_Lost' Picture.
If you need to get the stream interrupted when such cutting – re-connecting happens to the Inputs:
There is a simple web- operating shortcut for this:
We play the stream with VLC for testing as example.
The stream will disconnect automatically once remove the HDMI signal when we run:
http://ip/set_sys?kick_all=1 where IP is the one of the encoder of course.

TELNET:

The unit comes with Telnet-Port open. Some Admins do not want to be able to ping the port 23 and get an answer.
Anyway, the username and password are secret but Telnet can be completely disabled by manipulating the run -file in the OS in the unit. Please ask us for a Telnet-free Version if needed.

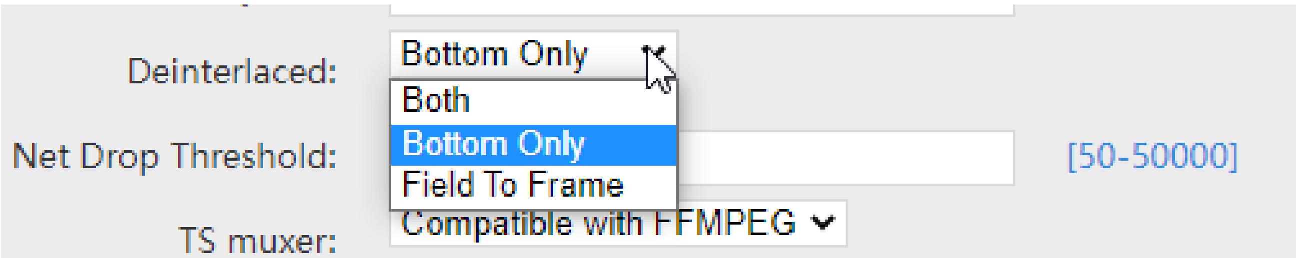
Traffic on the Network:

If you have heavy Traffic on your Network-Port because the encoder is connected to a heavy used network switch w/o filtering and no VLANs… it might be that the web-interface might get stuck or needs a long time to be reached and the reactions are very slow:
Decrease the NET DROP Threshold Value eg. From default 5000 to 500 – Trial and Error….



Also, a trick for HDMI / SDI / CVBS encoders:
If you face problems when encoding **and the 'Camera' picture moves right and left and the picture is not that smooth,** try to set the De-Interlaced mode as following:

SYSTEM- Advanced, submenu: please set the encoding **to Bottom Only,** then eventually need to **reboot** the encoder (dep. on Model):

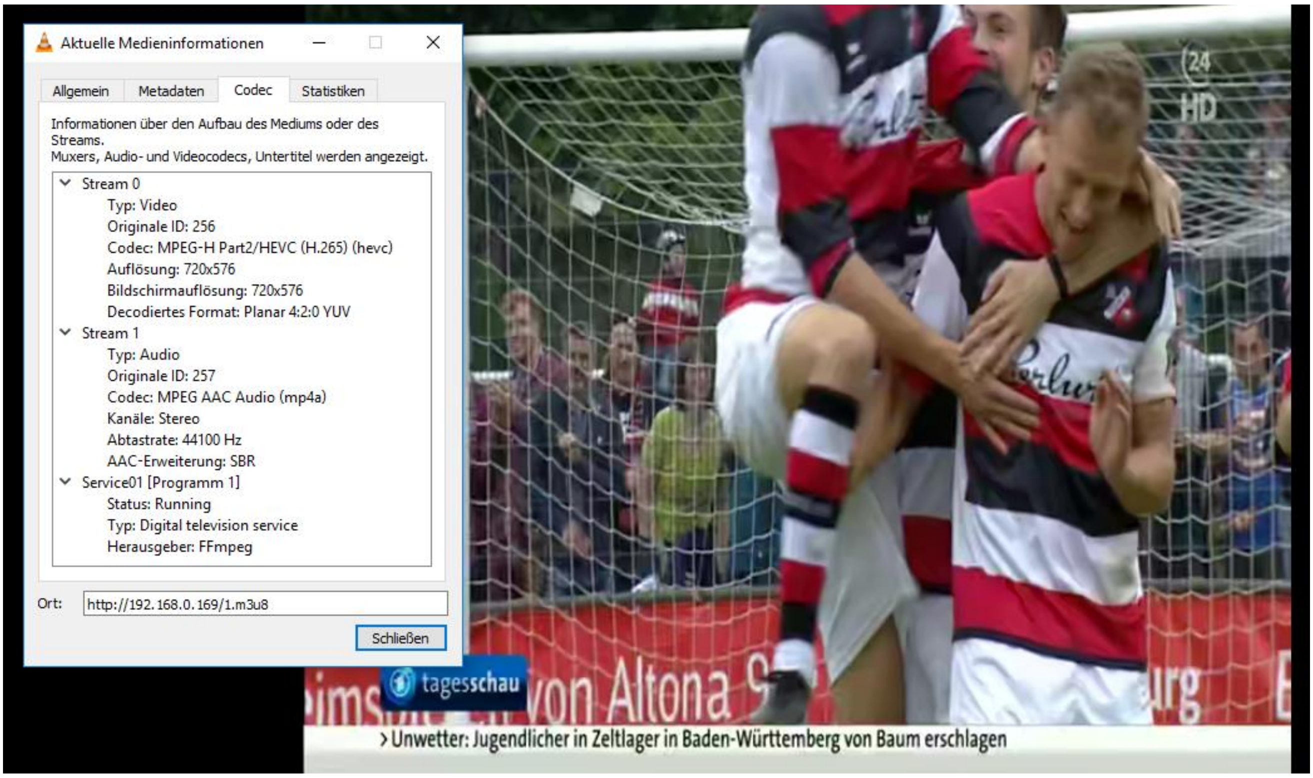


Because the Unicast pull the stream from the encoder like SRT or RTSP the receivers can only obtain a stream once from the same IP address or through VPNs… sometimes but
The new released http-based .mp4 streaming (new hde-265L) as address can be used for - or is limited to 5 connections …
- Remark: VPN transmissions need to set both Transmitter and Receiver into the same Subnet to use Unicast RTSP!
RTMP(s) see chapters VIMEO and Youtube examples

Parallel reception of Unicasts:

We tested a Full HD Encoder working with on parallel reception > 13 devices received the streams in unicast with parallel different protocols enabled from the same encoder… so, the limit will be reached of course but we did not finally test.

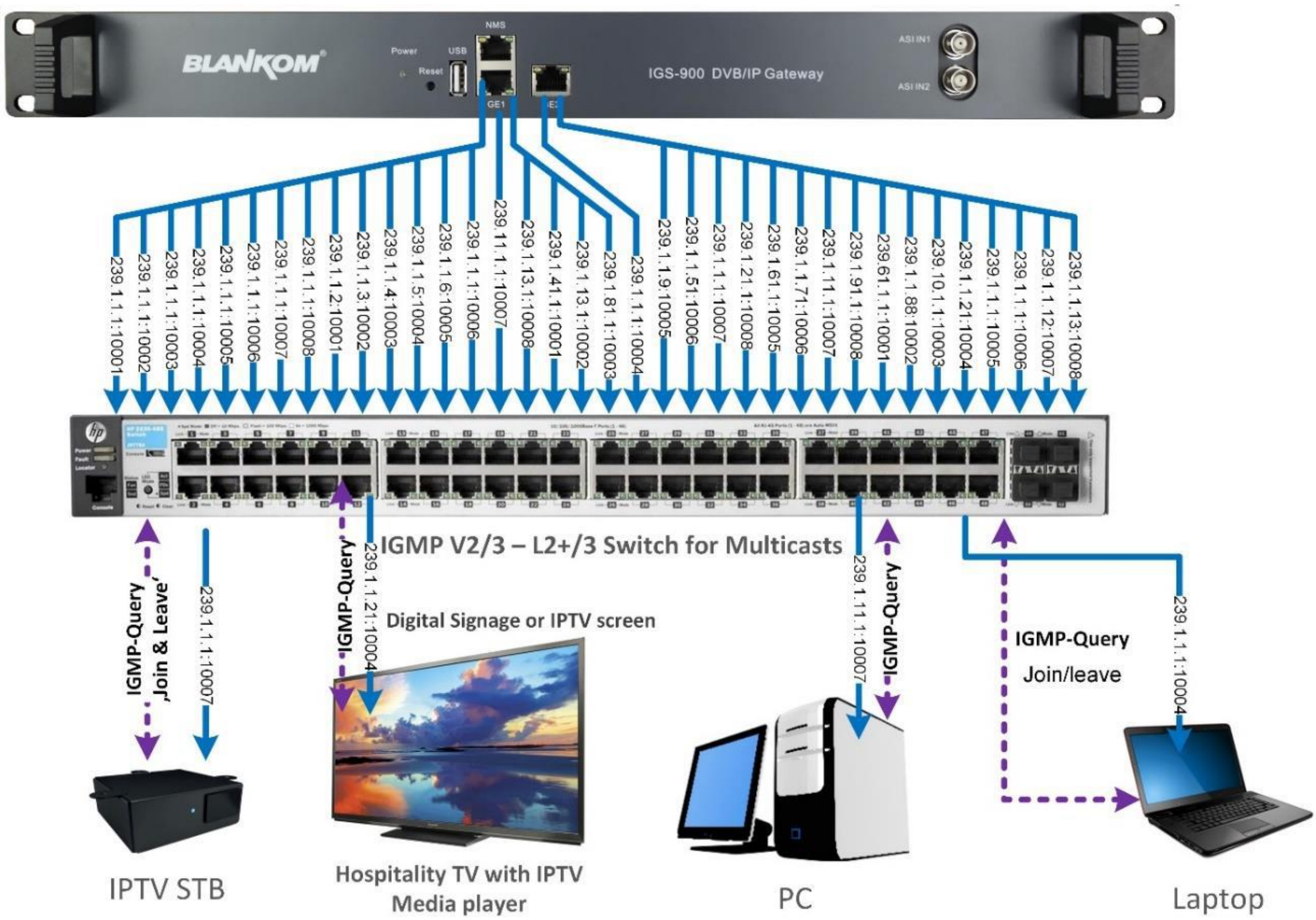
Please assure the corresponding picture settings if downscaling i.e., from 16:9 to 5:4 would squeeze the picture:



IGMP in Multicast Streaming Networks:

What is IGMP Querying

and IGMP Snooping and why would I need it on my network?
IGMP is a network layer (Layer 3) protocol used to establish membership in a Multicast group and can register a router to receive specific Multicast traffic. (Refer to RFC 1112 and RFC 2236 for information on IGMP versions 2 and 3). Multicast aware switches are slowly making their way into the network cores for businesses and universities that have heavy traffic to move through their networks. Multicast filtering is achieved by dynamic group control management. By default, all Multicast traffic should be blocked until requested by a Multicast group member. (Default behaviour depends on switch manufacturer.) **The master of the IGMP filter lists is the router or switch that is configured to act as the IGMP Query.** The responsibility of the Query is to send out IGMP group membership queries on a timed interval, to retrieve IGMP membership reports from active members, and to allow updating of the group membership tables. A ***Layer 2*** switch supporting IGMP Snooping can ***passively snoop*** on IGMP Query, Report, and Leave (IGMP version 2) packets transferred between IP Multicast routers/switches and IP Multicast hosts to determine the IP Multicast group membership. IGMP snooping checks IGMP packets passing through the network, picks out the group registration, and configures Multicasting accordingly.
See illustration:

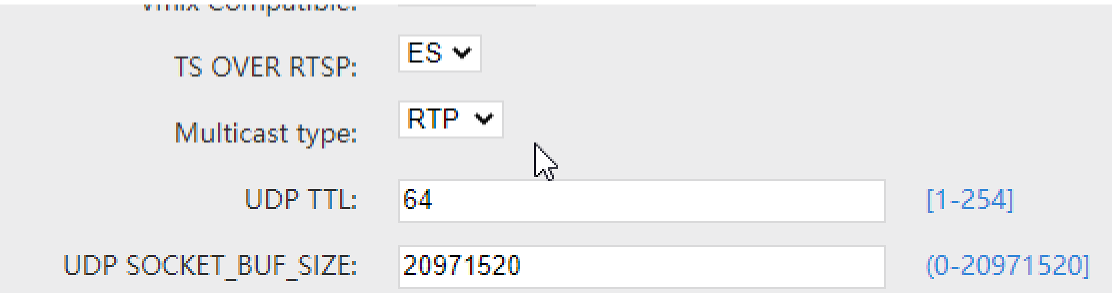


Without IGMP Querying/Snooping, Multicast traffic is treated in the same manner as a Broadcast transmission, which forwards packets to all ports on the network. With IGMP Querying/Snooping, Multicast traffic is only forwarded to ports that are members of that Multicast group. IGMP Snooping generates no additional network traffic, which significantly reduces the Multicast traffic passing through your switch.
If your network distribution core does not support IGMP Querying/Snooping, the AVN streams will still function as designed but your network may be subjected to high traffic loads and condensed collision domain due to the broadcasting action used by the older switch or hub. If this is the case, you may wish to isolate the streaming nodes within the network so that the streams may be viewed without crossing the normal network traffic along its path.

Recommendation: Not only Snooping but IGMP V2 or V3 switches with Layer2+ (the + stand for extra features like IGMP full support) so better Layer 3 is the best solution.

So, if you use Multicast UDP or RTP streaming

And:

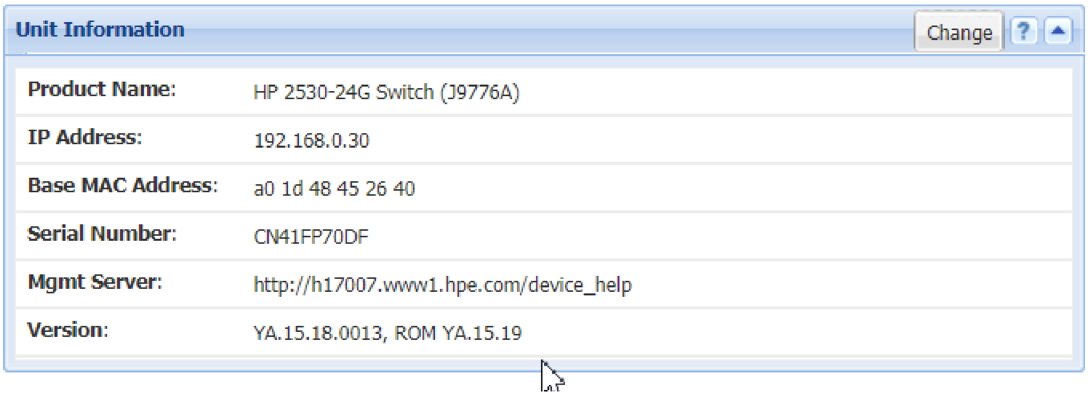


ends up in:

You should take care about IGMP in your Switch not to flood the whole network with these multicasts.

Suggestion: CAT 6E Ethernet cable for Gigabit Ethernet, DSTP (double shielded twisted pair) for the streaming ports.

As a **Multicast capable Switch** we recommend is the HP (ARUVA) 2530 24G or 48G.
(For Floor switches we have an own branded one and support IGMP as well) IGMP should be set to ON in the port configs. The latest HP Firmware might not be the best choice. Better to test IGMP functions before installation into a HOT running System and eventually do a downgrade of the Firmware. This one work:



Unit Information Change

Product Name:	HP 2530-24G Switch (J9776A)
IP Address:	192.168.0.30
Base MAC Address:	a0 1d 48 45 26 40
Serial Number:	CN41FP70DF
Mgmt Server:	http://h17007.www1.hpe.com/device_help
Version:	YA.15.18.0013, ROM YA.15.19

General notes about Streams:

Multicast streams:

Multicast Address Ranges:

We recommend, that the addressing of your Multicast streams should be in conjunction with this listings to avoid conflicts with other network equipment or protocols.
https://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

One small part from this:

IPv4 Multicast Address Space Registry

Last Updated
2018-01-05

Expert(s)
Stig Venaas

Note

Host Extensions for IP Multicasting [RFC1112] specifies the extensions required of a host implementation of the Internet Protocol (IP) to support multicasting. The multicast addresses are in the range 224.0.0.0 through 239.255.255.255. Address assignments are listed below.

The range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols, such as gateway discovery and group membership reporting. Multicast routers should not forward any multicast datagram with destination addresses in this range, regardless of its TTL.

Available Formats XML HTML Plain text
Registries included below

- Local Network Control Block (224.0.0.0 - 224.0.0.255 (224.0.0/24))
- Internetwork Control Block (224.0.1.0 - 224.0.1.255 (224.0.1/24))
- AD-HOC Block I (224.0.2.0 - 224.0.255.255)
- RESERVED (224.1.0.0-224.1.255.255 (224.1/16))
- SDP/SAP Block (224.2.0.0-224.2.255.255 (224.2/16))
- AD-HOC Block II (224.3.0.0-224.4.255.255 (224.3/16, 224.4/16))
- RESERVED (224.5.0.0-224.251.255.255 (251 /16s))
- DIS Transient Groups 224.252.0.0-224.255.255.255 (224.252/14))
- RESERVED (225.0.0.0-231.255.255.255 (7 /8s))
- Source-Specific Multicast Block (232.0.0.0-232.255.255.255 (232/8))
- GLOP Block
- AD-HOC Block III (233.252.0.0-233.255.255.255 (233.252/14))
- Unicast-Prefix-based IPv4 Multicast Addresses
- Scoped Multicast Ranges
- Relative Addresses used with Scoped Multicast Addresses

Multicast (as opposed to unicast) is used to send UDP packets from 1 source to multiple destination servers. This is useful for example for streaming from a satellite/DVB-T receiver to multiple receiving PCs for playback. Multicast can also be used on the output of an encoder to feed multiple streaming servers. Multicast only works with UDP and is not possible with TCP due to the 2 way nature of TCP, most commonly multicast is used with RTP and MPEG2-TS.

A multicast IP address must be chosen according to IANA information, we recommend using an address in the range **239.0.0.0 to 239.255.255.255** as this is reserved for private use. Using multicast addresses in the 224.0.0.0 range may clash with existing services and cause your stream to fail. For more details see http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml

Choosing a UDP port number for multicast streams is also important. Even if you use a different multicast IP for each of your streams, we strongly recommend using different UDP port numbers as well. This is because a server and all software running on the server receives ALL multicast traffic on an open port and extra processing is required to filter out the required traffic. If the each stream arrives on a different port, the server can safely ignore any traffic on ports that are not open. Port numbers MUST be chosen so that don't clash with any existing services or ephemeral ranges. The ephemeral range for Windows Vista, 7, 2008 is 49152 to 65535, for older Windows it is 1025 to 5000 and for Linux it is 32768 to 61000. For more information on Windows see http://support.microsoft.com/kb/929851 Care should also be taken to avoid system ports 0 to 1024. See http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml Generally one of the unassigned User Ports (**1024-49151**) should be used, you can run the ***netstat -abn*** (as admin under windows) command to see which ports are currently in use.

Registered port

A **registered port** is a network port (a sub-address defined within the Internet Protocol, in the range 1024–49151) assigned by the Internet Assigned Numbers Authority (IANA) (or by Internet Corporation for Assigned Names and Numbers (ICANN) before March 21, 2001,[1] or by USC/ISI before 1998) for use with a certain protocol or application.

Ports with numbers 0–1023 are called *system or well-known ports*; ports with numbers 1024-49151 are called *user or registered ports*, and ports with numbers 49152-65535 are called *dynamic and/or private ports*.[2] Both system and user ports are used by transport protocols (TCP, UDP, DCCP, SCTP) to indicate an application or service.

- **Ports 0–1023** – system or well-known ports
- **Ports 1024–49151** – user or registered ports
- **Ports >49151** – dynamic / private ports

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Range for Ephemeral port

The Internet Assigned Numbers Authority (IANA) suggests the range 49152 to 65535 ($2^{15}+2^{14}$ to $2^{16}-1$) for dynamic or private ports.[1]

Many Linux kernels use the port range 32768 to 61000.[note 2] FreeBSD has used the IANA port range since release 4.6. Previous versions, including the Berkeley Software Distribution (BSD), use ports 1024 to 5000 as ephemeral ports.[2][3]

Microsoft Windows operating systems through XP use the range 1025–5000 as ephemeral ports by default.[4] Windows Vista, Windows 7, and Server 2008 use the IANA range by default.[5] Windows Server 2003 uses the range 1025–5000 by default, until Microsoft security update MS08-037 from 2008 is installed, after which it uses the IANA range by default.[6] Windows Server 2008 with Exchange Server 2007 installed has a default port range of 1025–60000.[7] In addition to the default range, all versions of Windows since Windows 2000 have the option of specifying a custom range anywhere within 1025–65535.[8][9]

Packet structure

UDP Header

Offsets	**Octet**	**0**	**1**	**2**	**3**
Octet	**Bit**	**0 1 2 3 4 5 6 7**	**8 9 10 11 12 13 14 15**	**16 17 18 19 20 21 22 23**	**24 25 26 27 28 29 30 31**
0	**0**	Source port		Destination port	
4	**32**	Length		Checksum	

The UDP header consists of 4 fields, each of which is 2 bytes (16 bits).[1] The use of the fields "Checksum" and "Source port" is optional in IPv4 (pink background in table). In IPv6 only the source port is optional (see below).

Source port number

This field identifies the sender's port when meaningful and should be assumed to be the port to reply to if needed. If not used, then it should be zero. If the source host is the client, the port number is likely to be an ephemeral port number. If the source host is the server, the port number is likely to be a well-known port number.[4]

Destination port number

This field identifies the receiver's port and is required. Similar to source port number, if the client is the destination host then the port number will likely be an ephemeral port number and if the destination host is the server then the port number will likely be a well-known port number.[4]

Length

A field that specifies the length in bytes of the UDP header and UDP data. The minimum length is 8 bytes because that is the length of the header. The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram. However the actual limit for the data length, which is imposed by the underlying IPv4 protocol, is 65,507 bytes (65,535 − 8 byte UDP header − 20 byte IP header).[4]

In IPv6 jumbograms it is possible to have UDP packets of size greater than 65,535 bytes.[5] RFC 2675 specifies that the length field is set to zero if the length of the UDP header plus UDP data is greater than 65,535.

Checksum

The checksum field may be used for error-checking of the header and data. This field is optional in IPv4, and mandatory in IPv6.[6] The field carries all-zeros if unused.[7]

RTP:

apart from: https://tools.ietf.org/html/rfc3550

Chapter 11:

RTP relies on the underlying protocol(s) to provide demultiplexing of RTP data and RTCP control streams. For UDP and similar protocols, RTP SHOULD use an **even** destination port number and the corresponding RTCP stream SHOULD use the next higher (odd) destination port number.

For applications that take a single port number as a parameter and derive the RTP and RTCP port pair from that number, if an odd number is supplied then the application SHOULD replace that number with the next lower (even) number to use as the base of the port pair. For applications in which the RTP and RTCP destination port numbers are specified via explicit, separate parameters (using a signalling protocol or other means), the application MAY disregard the restrictions that the port numbers be even/odd and consecutive although the use of an even/odd port pair is still encouraged. The RTP and RTCP port numbers MUST NOT be the same since RTP relies on the port numbers to demultiplex the RTP data and RTCP control streams.

In a unicast session, both participants need to identify a port pair for receiving RTP and RTCP packets. Both participants MAY use the same port pair. A participant MUST NOT assume that the source port of the incoming RTP or RTCP packet can be used as the destination port for outgoing RTP or RTCP packets. When RTP data packets are being sent in both directions, each participant's RTCP SR packets MUST be sent to the port that the other participant has specified for reception of RTCP. The RTCP SR packets combine sender information for the outgoing data plus reception report information for the incoming data. If a side is not actively sending data (see Section 6.4), an RTCP RR packet is sent instead.

RTP (Real-Time Transport Protocol)	
Familie:	Netzwerkprotokoll
Einsatzgebiet:	Transport von Medien-Streams
Port:	beliebiger freier, gerader Port größer 1024
RTP im TCP/IP-Protokollstapel:	
Anwendung	**RTP**
Transport	UDP
Internet	IP (IPv4, IPv6)
Netzzugang	Ethernet · Token Bus · Token Ring · FDDI · …
Standard:	RFC 3550 (RTP: A Transport Protocol for Real-Time Applications, 2003)

any port (**even**, not odd > 1024)

Appendix B: ONVIF audio and video playback specification

1. RTSP usage
The replay protocol is based on RTSP [RFC 2326]. However, because RTSP does not directly support many of the features required by CCTV applications, this standard defines several extensions to the protocol; these are detailed below.

This standard makes the following stipulations on the usage of RTSP:

1. RTP/RTSP/HTTP/TCP shall be supported by the server. This is the same transport protocol as a device that implements media streaming through the media service shall support, and the same requirements shall apply to replay streaming.
2. The server shall support the unicast RTP/UDP transport for streaming.
3. Clients should use a TCP-based transport for replay, in order to achieve reliable delivery of media packets.
4. The server MAY elect not to send RTCP packets during replay. In typical usage RTCP packets are not required, because usually a reliable transport will be used, and because absolute time information is sent within the stream, making the timing information in RTCP sender reports redundant.

2. RTSP describe
The SDP returned by the RTSP describe command shall include the Track Reference for each track of the recording to allow a client to map the tracks presented in the SDP to tracks of the recording. The tag shall use the following format:

a:x-onvif-track:<TrackReference>

For example:

```
NVS - NVT:  DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                  Cseq: 1
                  User-Agent: ONVIF Rtsp client
                  Accept: application/sdp

NVT - NVS:  RTSP/1.0 200 OK
                  Cseq: 1
                  Content-Type: application/sdp
                  Content-Length: xxx

v=0
                  o= 2890842807 IN IP4 192.168.0.1
                  m=video 0 RTP/AVP 26
                        a=control:rtsp://192.168.0.1/video
                  a=x-onvif-track:VIDEO001
                  m=audio 0 RTP/AVP 98
                  a=control:rtsp://192.168.0.1/audio
                  a=x-onvif-track:AUDIO001
```

3. RTP header extension
In order to allow clients to report a stable and accurate timestamp for each frame played back regardless of the direction of playback, it is necessary to associate an absolute timestamp with each packet, or each group of packets with the same RTP timestamp (e.g. a video frame). This is achieved using an RTP header extension containing an NTP timestamp and some additional information also useful for replay.
The replay mechanism uses the extension ID 0xABAC for the replay extension.
Below shows the general form of an RTP packet containing this extension:

V=2	P	X=1	CC	M	PT	sequence number
timestamp						
synchronization source (SSRC) identifier						
0xABAC						length=3
NTP timestamp…						
...NTP timestamp						
C	E	D	T	mbz	Cseq	padding
payload…						

The fields of this extension are as follows:

• NTP timestamp. An NTP [RFC 1305] timestamp indicating the absolute UTC time associated with the access unit.
• C: 1 bit. Indicates that this access unit is a synchronization point or "clean point", e.g. the start of an intra-coded frame in the case of video streams.
• E: 1 bit. Indicates the end of a contiguous section of recording. The last access unit in each track before a recording gap, or at the end of available footage, shall have this bit set. When replaying in reverse, the E flag shall be set on the last frame at the end of the contiguous section of recording.
• D: 1 bit. Indicates that this access unit follows a discontinuity in transmission. It is primarily used during reverse replay; the first packet of each GOP has the D bit set since it does not chronologically follow the previous packet in the data stream
• T: 1 bit. Indicates that this is the terminal frame on playback of a track. A device should signal this flag in both forward and reverse playback whenever no more data is available for a track.
• mbz: This field is reserved for future use and must be zero.
• Cseq: 1 byte. This is the low-order byte of the Cseq value used in the RTSP PLAY command that was used to initiate transmission. When a client sends multiple, consecutive PLAY commands, this value may be used to determine where the data from each new PLAY command begins.

The replay header extension shall be present in the first packet of every access unit (e.g. video frame).

3.1 NTP Timestamps
The NTP timestamps in the RTP extension header shall correspond to the wallclock time as measured at the original frame grabber before encoding of the stream.
For forward playback of I and P frames the NTP timestamps in the RTP extension header shall increase monotonically over successive packets within a single RTP stream.

3.2 Compatibility with the JPEG header extension
The replay header extension may co-exist with the header extension used by the JPEG RTP profile; this is necessary to allow replay of JPEG streams that use this extension. The JPEG extension is simply appended to the replay extension; its presence is indicated by an RTP header extension length field with a value greater than 3, and by the extension start codes of 0xFFD8 or 0xFFFF at the start of the fourth word of the extension content.

The following illustrates a JPEG packet that uses both extensions:

<table>
<tr><td>V=
2</td><td>P</td><td>X=
1</td><td>CC</td><td>M</td><td>PT</td><td>sequence number</td></tr>
<tr><td colspan="7">timestamp</td></tr>
<tr><td colspan="7">synchronization source (SSRC) identifier</td></tr>
<tr><td colspan="6">0xABAC</td><td>length=N+4</td></tr>
<tr><td colspan="7">NTP timestamp…</td></tr>
<tr><td colspan="7">...NTP timestamp</td></tr>
<tr><td>C</td><td>E</td><td>D</td><td colspan="2">mbz</td><td>Cseq</td><td>padding</td></tr>
<tr><td colspan="6">0xFFD8</td><td>jpeglength=N</td></tr>
<tr><td colspan="7">extension payload: sequence of additional JPEG marker segments padded with 0xFF to the total extension length</td></tr>
<tr><td colspan="7">payload…</td></tr>
</table>

4. RTSP Feature Tag

The Replay Service uses the "onvif-replay" feature tag to indicate that it supports the RTSP extensions described in this standard. This allows clients to query the server's support for these extensions using the Require header as described in [RFC 2326] section 5.3.1.

Example:

```
C->S:   SETUP rtsp://server.com/foo/bar/baz.rm RTSP/1.0
        Cseq: 302
        Require: onvif-replay

S->C:   RTSP/1.0 551 Option not supported
        Cseq: 302
        Unsupported: onvif-replay
```

The Replay Server shall accept a SETUP and PLAY command that includes a Require header containing the onvif-replay feature tag.

5. Initiating Playback

Playback is initiated by means of the RTSP PLAY method. For example:

```
PLAY rtsp://192.168.0.1/path/to/recording RTSP/1.0
Cseq: 123
Session: 12345678
Require: onvif-replay
Range: clock=20090615T114900.440Z-
Rate-Control: no
```

The ReversePlayback capability defined in the ONVIF Replay Control Service Specification signals if a device supports reverse playback. Reverse playback is indicated using the Scale header field with a negative value. For example to play in reverse without no data loss a value of ̃10 would be used.

```
PLAY rtsp://192.168.0.1/path/to/recording RTSP/1.0
Cseq: 123
Session: 12345678
Require: onvif-replay
Range: clock=20090615T114900.440Z-
Rate-Control: no
Scale: -1.0
```

If a device supports reverse playback it shall accept a Scale header with a value of -1.0. A device MAY accept other values for the Scale parameter. Unless the Rate-Control header is set to "no" (see below), the Scale parameter is used in the manner described in [RFC 2326]. If Rate-Control is set to "no", the Scale parameter, if it is present, shall be either 1.0 or -1.0, to indicate forward or reverse playback respectively. If it is not present, forward playback is assumed.

5.1 Range header field

A device shall support the Range field expressed using absolute times as defined by [RFC 2326]. Absolute times are expressed using the utc-range from [RFC 2326].
Either open or closed ranges may be used. In the case of a closed range, the range is increasing (end time later than start time) for forward playback and decreasing for reverse playback. The direction of the range shall correspond to the value of the Scale header.
In all cases, the first point of the range indicates the starting point for replay
The time itself shall be given as

```
utc-range = "clock" ["=" utc-range-spec]
utc-range-spec = ( utc-time "-" [ utc-time ] ) / ( "-" utc-time )
utc-time = utc-date "T" utc-clock "Z"
utc-date = 8DIGIT
utc-clock = 6DIGIT [ "." 1*9DIGIT ]
```

as defined in [RFC2326].

Examples:

```
PLAY rtsp://192.168.0.1/path/to/recording RTSP/1.0
Cseq: 123
Session: 12345678
Require: onvif-replay
Range: clock=20090615T114900.440Z-20090615T115000Z
Rate-Control: no
```

```
PLAY rtsp://192.168.0.1/path/to/recording RTSP/1.0
Cseq: 123
Session: 12345678
Require: onvif-replay
Range: clock=20090615T115000.440Z-20090615T114900Z
Rate-Control: no
Scale: -1.0
```

5.2 Rate-Control header field

This specification introduces the Rate-Control header field, which may be either "yes" or "no". If the field is not present, "yes" is assumed, and the stream is delivered in real time using standard RTP timing mechanisms. If this field is "no", the stream is delivered as fast as possible, using only the flow control provided by the transport to limit the delivery rate.
The important difference between these two modes is that with "Rate-Control=yes", the server is in control of the playback speed, whereas with "Rate-Control=no" the client is in control of the playback speed. Rate-controlled replay will typically only be used by non-ONVIF specific clients as they will not specify "Rate-Control=no".
When replaying multiple tracks of a single recording, started by a single RTSP PLAY command and not using rate-control, the data from the tracks should be multiplexed in time in the same order as they were recorded.
An ONVIF compliant RTSP server shall support operation with "Rate-Control=no" for playback.

5.3 Frames header field

The Frames header field may be used to reduce the number of frames that are transmitted, for example to lower bandwidth or processing load. Three modes are possible:

1. Intra frames only. This is indicated using the value "intra", optionally followed by a minimum interval between successive intra frames in the stream. The latter can be used to limit the number of frames received even in the presence of "I-frame storms" caused by many receivers requesting frequent I-frames.
2. Intra frames and predicted frames only. This is indicated using the value "predicted". This value can be used to eliminate B-frames if the stream includes them.

3. All frames. This is the default.

Examples:

To request intra frames only:
Frames: intra

To request intra frames with a minimum interval of 4000 milliseconds:
Frames: intra/4000

To request intra frames and predicted frames only:
Frames: predicted

To request all frames (note that it is not necessary to explicitly specify this mode but the example is included for completeness):
Frames: all

The interval argument used with the "intra" option refers to the recording timeline, not playback time; thus for any given interval the same frames are played regardless of playback speed. The interval argument shall NOT be present unless the Frames option is "intra".
The server shall support the Frames header field. This does not preclude the use of the Scale header field as an alternative means of limiting the data rate. The implementation of the Scale header field may vary between different server implementations, as stated by [RFC 2326].
An ONVIF compliant RTSP server shall support the Frames parameters "intra" and "all" for playback.

5.4 Synchronization points
The transmitted video stream shall begin at a synchronization point (see section "Synchronization Point" of the ONVIF Media Service Specification). The rules for choosing the starting frame are as follows:

- If the requested start time is within a section of recorded footage, the stream starts with the first clean point at or before the requested start time. This is the case regardless of playback direction.
- If the requested start time is within a gap in recorded footage and playback is being initiated in the forwards direction, the stream starts with the first clean point in the section following the requested start time.
- If the requested start time is within a gap in recorded footage and playback is being initiated in the reverse direction, the stream starts with the last clean point in the section preceding the requested start time.

Contact:

Änderungen vorbehalten / Subject to change w/o notifications

IRENIS GmbH
Hauptstr. 29
D-31171 Nordstemmen - Germany

Web: www. blankom. de **E-Mail:** info @ blankom.de
Phone: +49 5069 4809781

Managing Director: Dipl. Ing. Murad Önol
Commercial Register: HRB 206370 / District Court Hildesheim

WEEE: DE 54333499