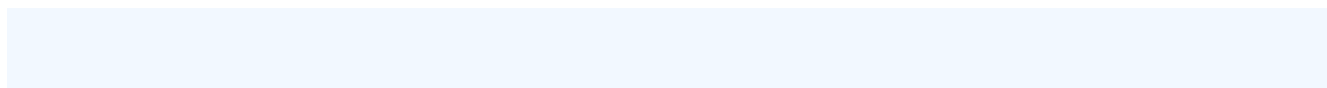


Video Processing Project

Project Web Hosting
Open Source Software

Please drop the [Video Processing Project](#) team a note if you use or find any of the offerings here useful. This will help us to determine if our contribution to the community is worthwhile.



Project Information

History

About this project:

This is the **Video Processing Project**.

This project was registered on SourceForge.net on Mar 23, 2009, and is described by the project team as follows:

Video processing source code for algorithms and tools used in software media pipelines (e.g. image scalers, colour converters, etc.) The currently available source code is written in C++ with their associated libraries and DirectShow™ Filters.

Background

This project was originally started by developers at the Council for Scientific and Industrial Research (CSIR) in South Africa as a mechanism for source code quality control. These functions and filters are used on various publicly funded projects and we felt that they might also be useful to the open source community as a whole. Your use of them and your comments or bug finds are a great help to us to improve the quality of our projects.

After 2020 the original CSIR seems to be dead. This set of filters has been adopted in Digiteq Automotive to provide some filters that could be used together with FG4 frame grabber - <https://fg4.digiteqautomotive.com/> . Support for RGB32 format has been added to all filters. All filters have newly added *IPersistStream* interface that allows to access to properties from any 3rd party application.

Please drop us a note if you use or find any of the offerings here useful. This will help us to determine if our contribution to the community is worthwhile.
Our broad objective with the style of coding used is to balance the speed of execution with readable code that is easy to maintain. We have tried to hide our implementations behind well defined interfaces to promote system decoupling.

Filters

These are a set of general filters for scaling, cropping, colour conversion, etc. that are completed items and ready for use. The implementations provided are designed for use with Microsoft's Direct Show™ framework and we have supplied some useful DirectShow™ base classes that further simplify the

development effort.

The following filters are installed when running the DQ_VPP_DshowFiltersInstaller-1.7.27.exe download:

- [Audio mixing filter](#)
- Compare filter
- [Crop filter](#)
- [Framerate display filter](#)
- [H264 encoder filter](#)
- [H264 decoder filter](#)
- [H264 source filter](#)
- [Picture in picture filter](#)
- [RGB 2 YUV converter](#)
- [Rotate filter](#)
- [RTSP source filter](#)
- [Scale filter](#)
- Tee filter
- [Timestamplogging filter](#)
- Toggle filter
- Utility filter
- [Video mixing filter](#)
- Virtual mic
- [YUV source filter](#)
- [YUV 2 RGB converter](#)

These filters can also be downloaded individually and registered using the regsvr32 command. In Graphedit they will then appear in the DirectShow™ category with the prefix "DQ VPP". The filters can be configured using their property pages. The full source code is available in the subversion repository.

We encourage community contributions for faster implementations that sit behind the same interfaces.

Video file formats

RGB555 - <https://learn.microsoft.com/en-us/windows/win32/directshow/uncompressed-rgb-video-subtypes>

High-order byte: Low-order byte:

X R R R R G G G G G B B B B

X = Don't care, R = Red, G = Green, B = Blue

RGB565

High-order byte: Low-order byte:

R R R R R G G G G G G B B B B

YUYV and YUY2

Both formats have a same contents

The term U is equivalent to Cb, and the term V is equivalent to Cr.

4:2:2 means 2:1 horizontal downsampling, with no vertical downsampling. Every scan line contains four Y samples for every two U or V samples.

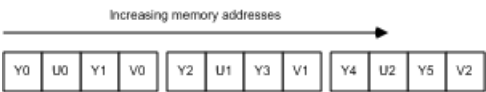
In YUY2 format, the data can be treated as an array of unsigned **char** values, where the first byte contains the first Y sample, the second byte contains the first U (Cb) sample, the third byte contains the second Y sample, and the fourth byte contains the first V (Cr) sample, as shown in the following diagram.

UYVY

This format is the same as the YUY2 format except the byte order is reversed—that is, the chroma and luma bytes are flipped

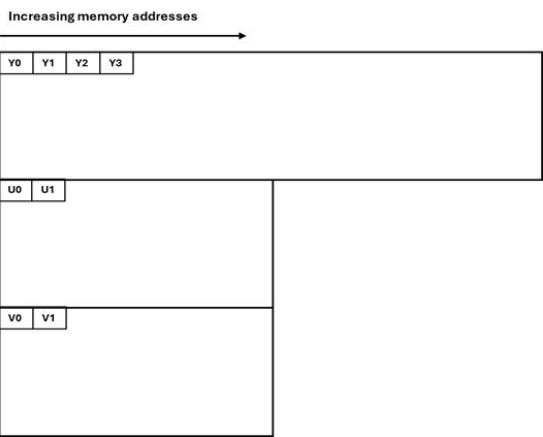


I422



In I422 format, all of the Y samples appear first in memory as an array of unsigned char values. This array is followed immediately by all of the U (Cb) samples.

The stride of the U plane is half the stride of the Y plane; and the U plane contains the same number of lines as the Y plane. The U plane is followed immediately by all of the V (Cr) samples, with the same stride and number of lines as the U plane, as shown in the following illustration. There is no padding for the Y/U/V planes.



I420, YUV420P_S, YV12 and IYUV

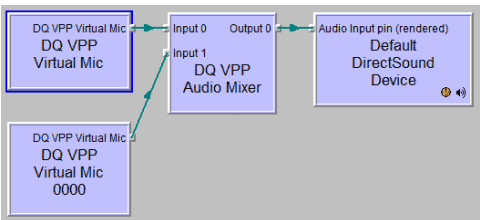
Y samples appear first in memory as an array of unsigned char values. This array is followed immediately by all of the U (Cb) samples. U and V planes have half rows and half columns.

Filter overview

Audio mixing filter

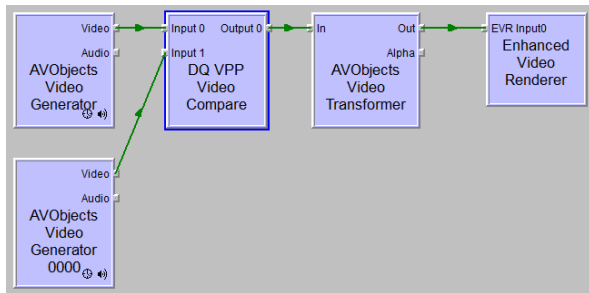
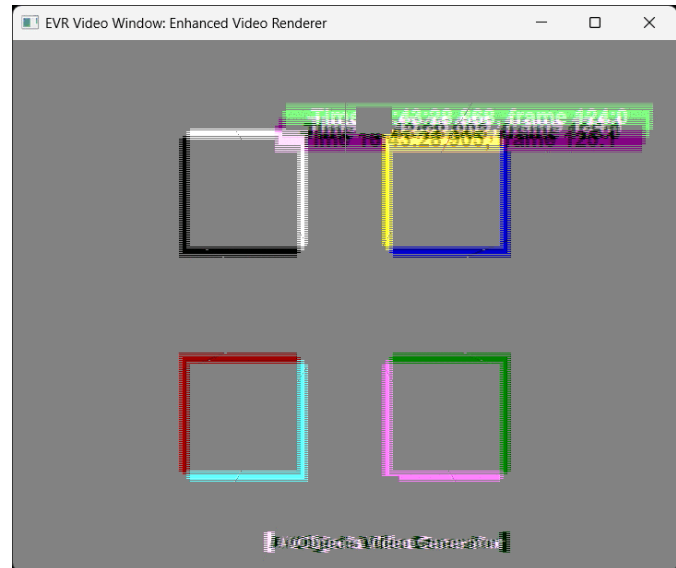
Sample filter that shows how multiple PCM streams can be mixed together.

Compare filter



Filter that compares 2 video signals and outputs their difference.

A main purpose is to verify whether 2 video signals are equivalent.

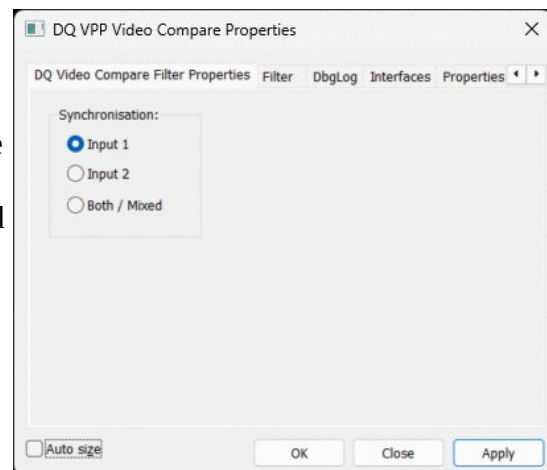


Input video formats supported:

- MEDIASUBTYPE_RGB24
- MEDIASUBTYPE_RGB32
- MEDIASUBTYPE_ARGB32
- MEDIASUBTYPE_YUY2; MEDIASUBTYPE_YUYV
- MEDIASUBTYPE_YVYU
- MEDIASUBTYPE_I420; MEDIASUBTYPE_YUV420P_S; MEDIASUBTYPE_YV1,
- MEDIASUBTYPE_IYUV
- MEDIASUBTYPE_UYVY, MEDIASUBTYPE_HDYC,
- MEDIASUBTYPE_AYUV

Output format must be the same as input format with one exception, formats in equivalent group like: (I420, YUV420P and YV12); (YUY2 and YUYV); (UYVY and HDYC) could be mixed.

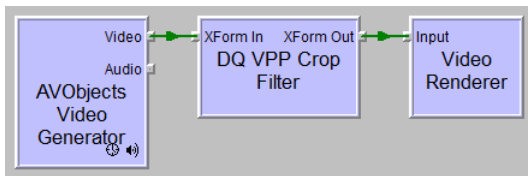
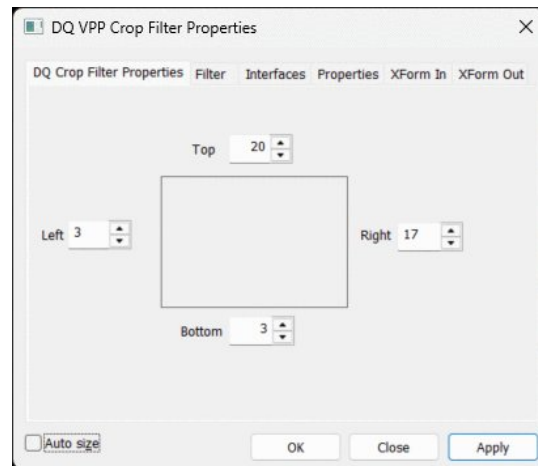
It is possible to choose frame generation on input 1 or input 2. Choosing “both” duplicates frame rate and it is not recommended.



Crop filter

Filter that crops RGB media.

User can choose how much pixels to cut from left, right top and bottom.



Supported videoformats:

1. MEDIASUBTYPE_ARGB32
2. MEDIASUBTYPE_RGB32
3. MEDIASUBTYPE_RGB24
4. MEDIASUBTYPE_AYUV
5. MEDIASUBTYPE_YV24
6. MEDIASUBTYPE_ARGB1555
7. MEDIASUBTYPE_ARGB4444
8. MEDIASUBTYPE_RGB555
9. MEDIASUBTYPE_RGB565
10. MEDIASUBTYPE_A2R10G10B10
11. MEDIASUBTYPE_A2B10G10R10
12. MEDIASUBTYPE_R16G16B16,
13. MEDIASUBTYPE_B16G16R16,
14. MEDIASUBTYPE_A16R16G16B16,
15. MEDIASUBTYPE_A16B16G16R16,
16. MEDIASUBTYPE_Y416,
17. MEDIASUBTYPE_Y410

Framerate display filter

Filter that estimates framerate and overlays it on the video using GDI.

H264 encoder filter

An H.264 encoder that implements the baseline profile. [-- More info on codec --](#) [-- More info on DirectShow filter --](#)

Please note that this filter is still not fully tested, it is not recommended for use.

H264 decoder filter

An H.264 decoder designed for use with the provided encoder or H264 source filter. The decoder uses a custom media type which can be configured on the encoder and source filters. [-- More info on codec](#)

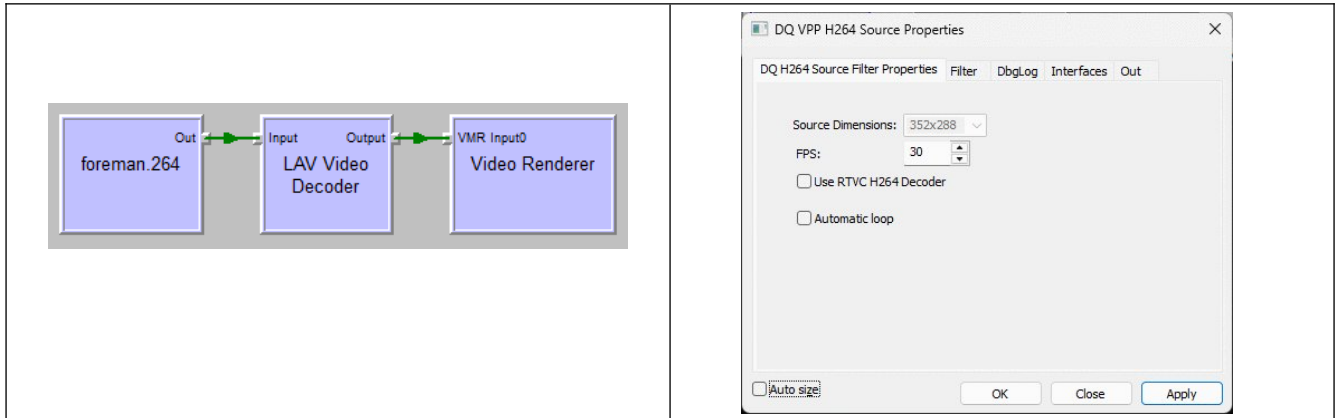
[-- -- More info on DirectShow filter --](#)

Please note that this filter is still not fully tested, it is not recommended for use.

H264 source filter

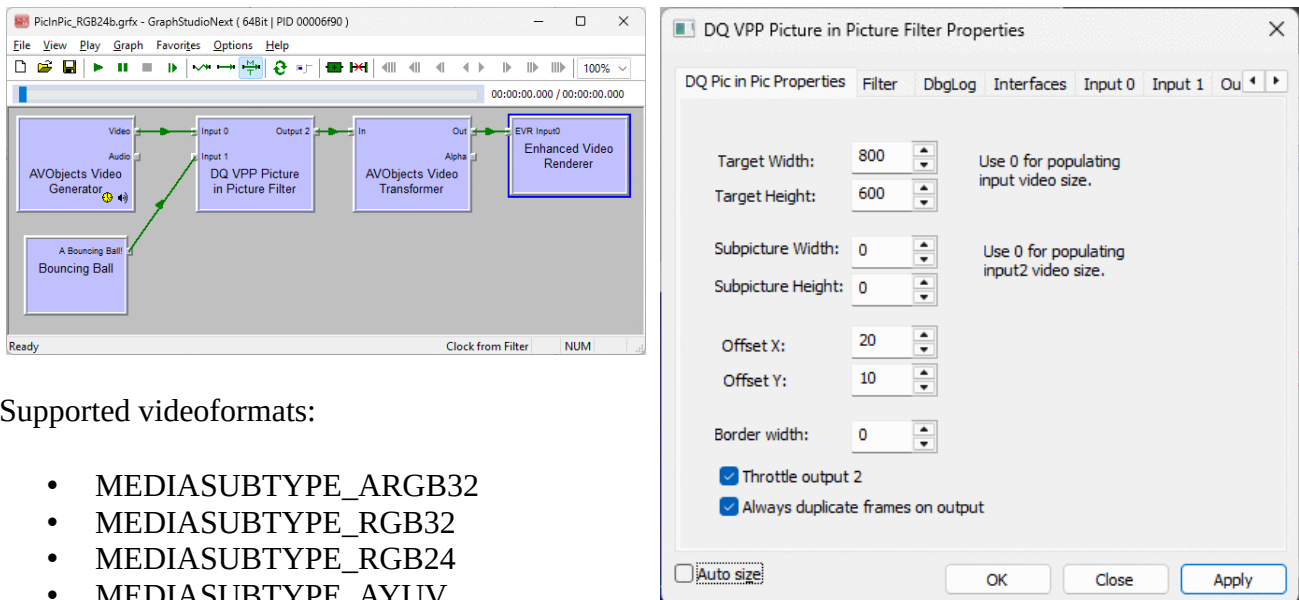
Source filter that can read raw .264 files. [Foreman sample file](#) [More info](#)

It is recommended to use LAV Video Decoder filter to decompress H264 video.



Picture in picture filter

This filter can include one video into another video. Filter optionally rescales included video.

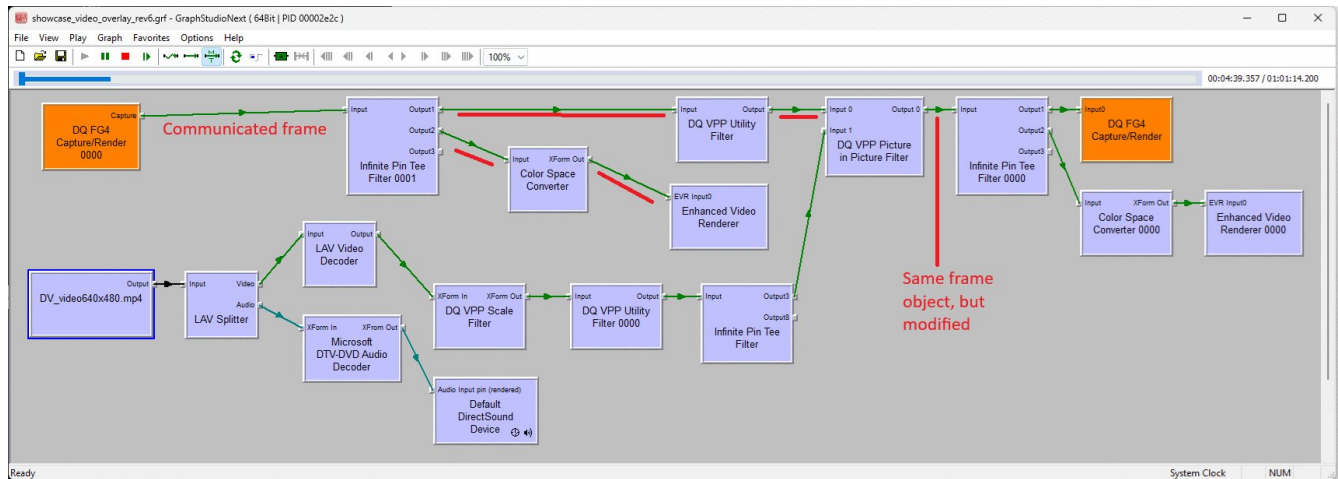


Supported videoformats:

- MEDIASUBTYPE_ARGB32
- MEDIASUBTYPE_RGB32
- MEDIASUBTYPE_RGB24
- MEDIASUBTYPE_AYUV
- MEDIASUBTYPE_YV24
- MEDIASUBTYPE_RGB555
- MEDIASUBTYPE_RGB565
- MEDIASUBTYPE_R16G16B16,
- MEDIASUBTYPE_B16G16R16,
- MEDIASUBTYPE_A16R16G16B16,

- MEDIASUBTYPE_A16B16G16R16

All pins must have the same videoformat.



Option – Always duplicate frames on output

Sometimes video with inserted picture could get back propagated. In this case please check option “Always duplicate frames on output”. This option enforces video data copy. If you do not use Infinite tee pin filter or something like this, option causes only CPU slowdown. This checkbox has no sense when you turn on scaling. In such a case frames must be cloned.

Option – Throttle output 2

If you use 2nd online video sources like camera or FG4 frame grabber, this option should be unticked. But if you decide to mix another video sourced from a file, throttling ensures that time is synchronized.

PicInPic filter implements optional video flip when nasty downstream filter needs negative height.

RGB 2 YUV converter

RGB24 to YUV420 planar converter. This filter can be connected to a dump filter to create raw YUV files. [More info](#)

This converter has only one property – Image invert.

Input video format

MEDIASUBTYPE_RGB24
MEDIASUBTYPE_RGB32
MEDIASUBTYPE_ARGB32
MEDIASUBTYPE_RGB555
MEDIASUBTYPE_RGB565

Output video format

MEDIASUBTYPE_AYUV
MEDIASUBTYPE_YV24
MEDIASUBTYPE_YUV420P_S
MEDIASUBTYPE_I420
MEDIASUBTYPE_YV12
MEDIASUBTYPE_IYUV
MEDIASUBTYPE_NV12
MEDIASUBTYPE_YUYV
MEDIASUBTYPE_YUY2
MEDIASUBTYPE_YVYU
MEDIASUBTYPE_UYVY

MEDIASUBTYPE_HDYC

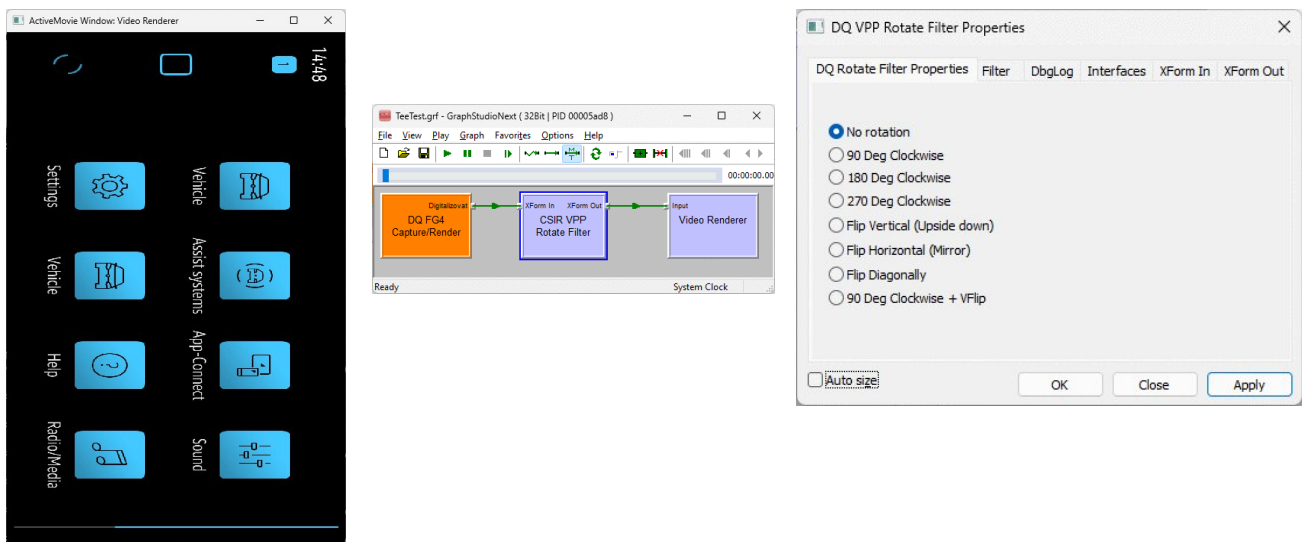
You can use any combination of input and output formats.

RTSP source filter

RTSP source filter that supports 8/16-bit PCM, AMR (via Monogram AMR filters), mp3 and H.264. This filter uses the liveMedia streaming library for RTSP/RTP/RTCP available at www.live555.com

Rotate filter

Filter that rotates video by 90, 180, 270 degrees, as well as mirroring vertically and horizontally. There exists 8 possible operations of this type, see filter property page.



Supported input video formats:

- MEDIASUBTYPE_RGB24
- MEDIASUBTYPE_RGB32
- MEDIASUBTYPE_ARGB32
- MEDIASUBTYPE_AYUV
- MEDIASUBTYPE_YV24
- MEDIASUBTYPE_RGB555
- MEDIASUBTYPE_RGB565
- MEDIASUBTYPE_R16G16B16,
- MEDIASUBTYPE_B16G16R16,
- MEDIASUBTYPE_A16R16G16B16,
- MEDIASUBTYPE_A16B16G16R16

Output format must be the same as input format.

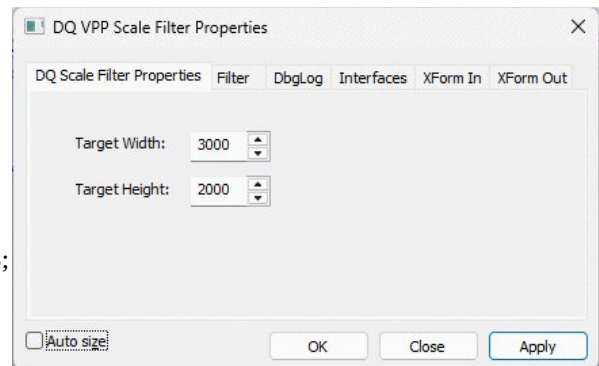
Scale filter

Scale filter that supports both upscaling and downscaling.

This filter has 2 properties width and height. When width and Height are set to 0, filter passes its input to output without resolution change.

Input video formats:

- MEDIASUBTYPE_RGB24
- MEDIASUBTYPE_RGB32
- MEDIASUBTYPE_ARGB32
- MEDIASUBTYPE_YUY2; MEDIASUBTYPE_YUYV
- MEDIASUBTYPE_YVYU
- MEDIASUBTYPE_I420; MEDIASUBTYPE_YUV420P_S;
MEDIASUBTYPE_IYUV
- MEDIASUBTYPE_UYVY, MEDIASUBTYPE_HDYC,
- MEDIASUBTYPE_AYUV
- MEDIASUBTYPE_YV24
- MEDIASUBTYPE_RGB555
- MEDIASUBTYPE_RGB565



Output format must be the same as input format with one exception: I420, YUV420P and YV12 could be mixed.

Scale filter implements optional video flip when nasty downstream filter needs negative height.

Tee filter

This filter has same functionality as infinite tee pin filter from Microsoft.

<https://learn.microsoft.com/en-us/windows/win32/directshow/infinite-pin-tee-filter>

This filter delivers samples delivered to its input pin to a variable number of output pins. When an instance of the filter is created, it has one output pin. Each time an output pin is connected, the filter creates another output pin. All output pins share the same media type as the input pin.

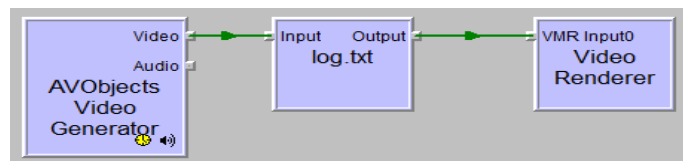
In some cases you can find DQ VPP Tee more convenient. If can optionally duplicate frame contents.

Timestamplogging filter

Filter that logs sample timestamps to file with current time. This is useful to calculate DirectShow filter sample processing times.

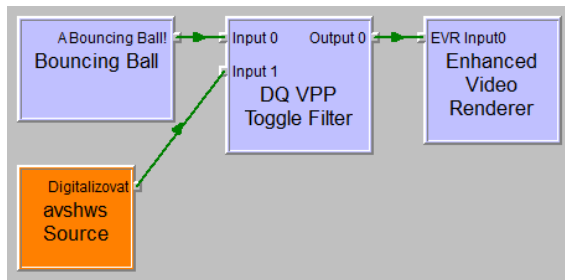
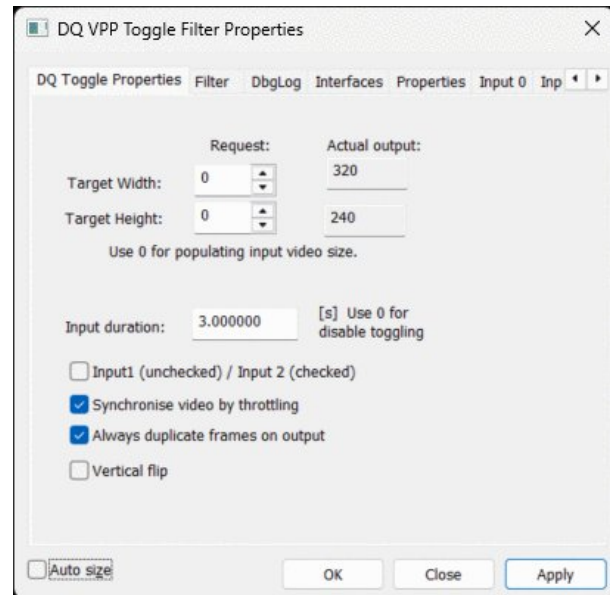
Sample output:

```
0 4000000 4000000 829440
4000000 8000000 4000000 829440
8000000 12000000 4000000 829440
12000000 16000000 4000000 829440
16000000 20000000 4000000 829440
20000000 24000000 4000000 829440
24000000 28000000 4000000 829440
28000000 32000000 4000000 829440
32000000 36000000 4000000 829440
36000000 40000000 4000000 829440
```



Toggle filter

Toggle filter allows to choose video from 2 different sources. Video can be periodically toggled.

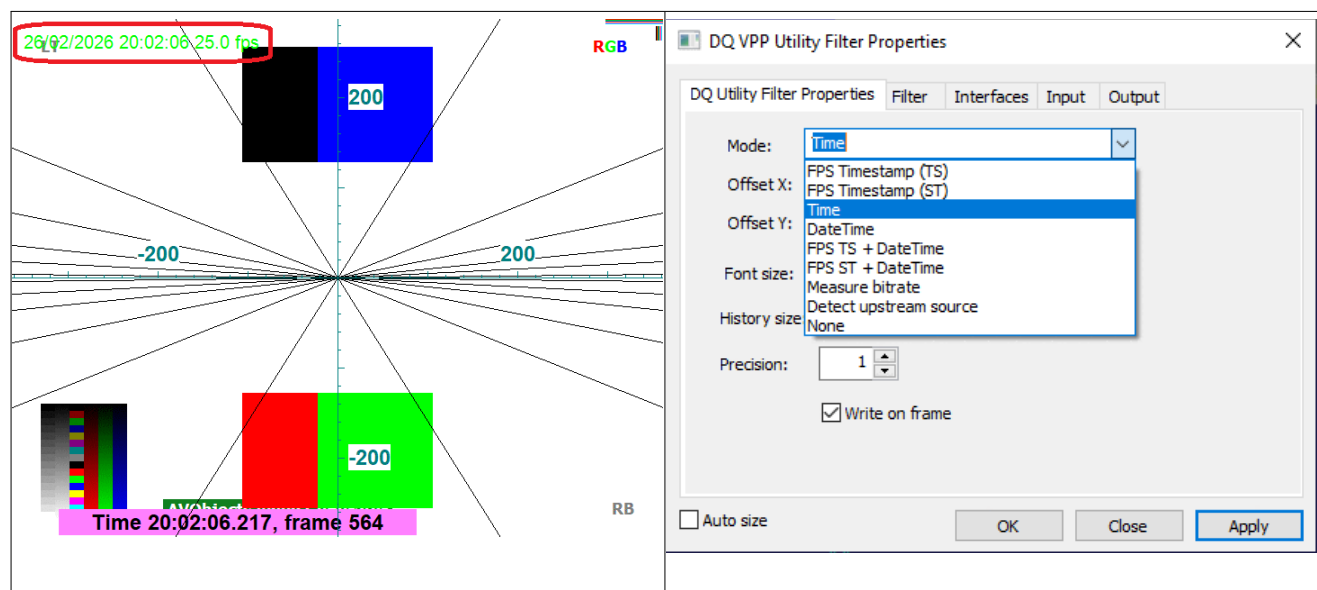


It is possible to run both video sources together. Normally it is possible to run either one video source or another video source.

This filter accepts any type of video on its input.

Utility filter

Utility filter inserts text watermarks into running video.

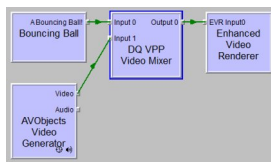
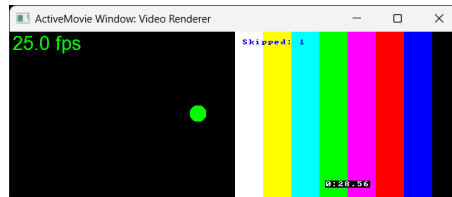
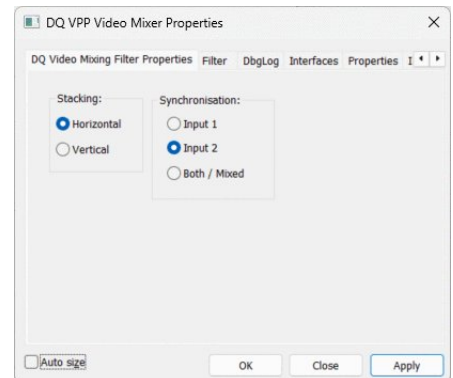


Video mixing filter

Sample video mixing filter that shows how multiple video streams can be mixed together.

Supported input video formats:

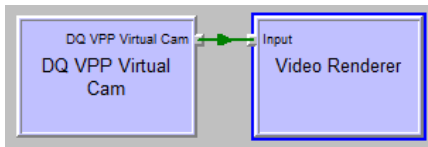
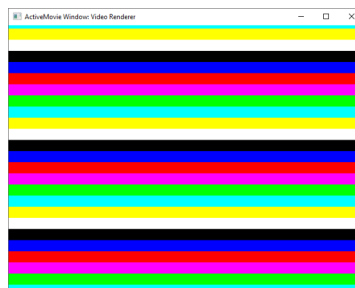
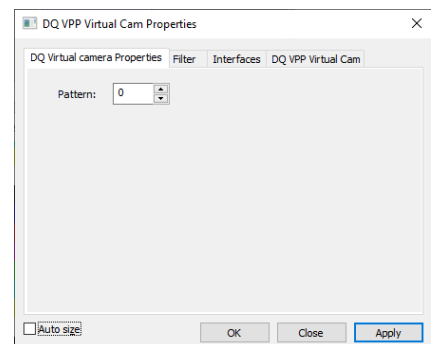
- MEDIASUBTYPE_RGB24
- MEDIASUBTYPE_RGB32
- MEDIASUBTYPE_ARGB32
- MEDIASUBTYPE_AYUV
- MEDIASUBTYPE_YV24



Video mixer can stack 2 videos horizontally or vertically.

Virtual Cam

Virtual cam is a simple video source.

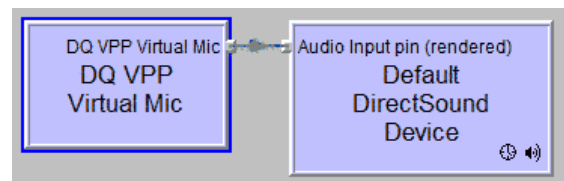


Virtual Mic

This filter generates test audio signal. It generates test beeps.

Supported output audio format:

- MEDIASUBTYPE_PCM



YUV source filter

Source filter for raw YUV media such as videos from the standard test sequences. [More info](#)

The raw video stream could be prepared with Microsoft's sample dump filter:
<https://learn.microsoft.com/en-us/windows/win32/directshow/dump-filter-sample>
 precompiled version could be found here:
<https://sourceforge.net/projects/direct-show-demo-filters/files/SampleFilters/>



There is one trick, that filename must contain video frame resolution like “video.720x576.YUV”. Character ‘x’ is used to separate width and height.

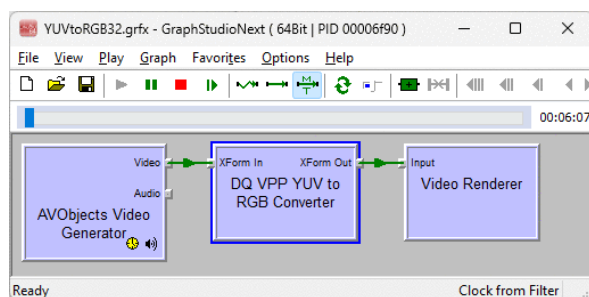
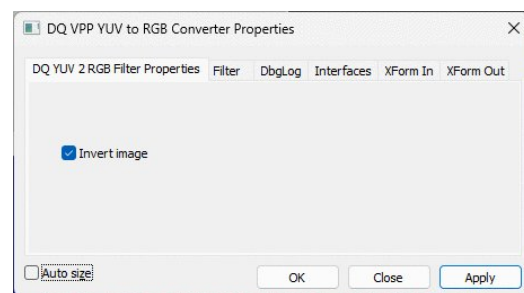
It normally uses I420 media video. When file name contains “444”, then video is handled like it contains AYUV media format.

There are another tricks, that substring “qcif” in a filename switches resolution “176x144” and “cif” in a filename switches resolution “352x288” – see also
<https://www.videoexpertsgroup.com/glossary/what-is-qcif>

YUV 2 RGB converter

Filter provides conversion from input to RGB based format. Can be used in conjunction with the YUV source filter. [More info](#)

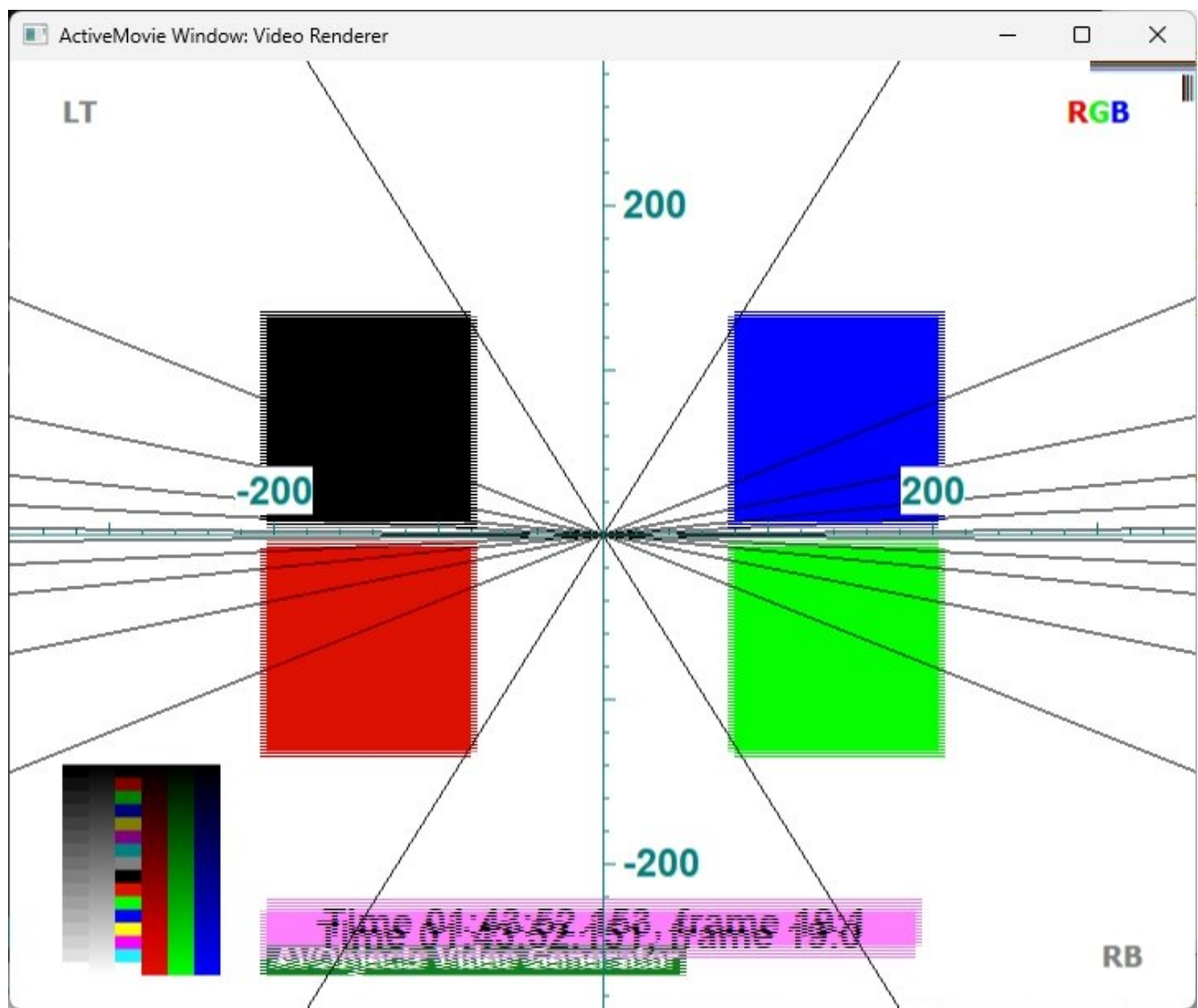
This converter outperforms original CSIR VPP YUV420toRGB and YUV444toRGB filters. These filters has been removed.



This converter has only one property – Image invert.

Input video format
 MEDIASUBTYPE_AYUV
 MEDIASUBTYPE_YV24
 MEDIASUBTYPE_YUV420P_S
 MEDIASUBTYPE_I420
 MEDIASUBTYPE_YV12
 MEDIASUBTYPE_IYUV
 MEDIASUBTYPE_NV12

Output video format
 MEDIASUBTYPE_RGB32
 MEDIASUBTYPE_ARGB32
 MEDIASUBTYPE_RGB24



MEDIASUBTYPE_YUYV
MEDIASUBTYPE_YUY2
MEDIASUBTYPE_YVYU
MEDIASUBTYPE_HDYC

You can use any combination of input and output formats.

Codecs

An H.264 video codec (encoding and decoding) is currently under development using a bottom-up

design approach.

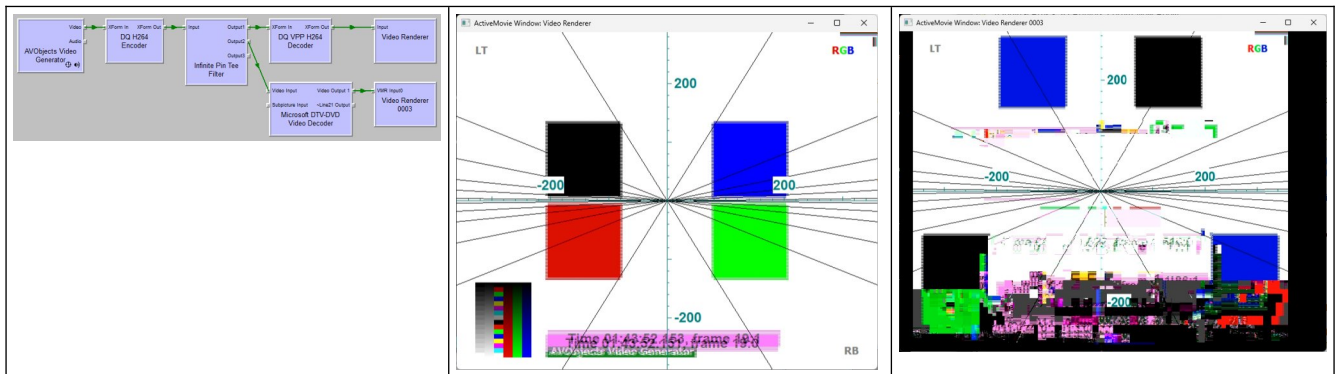
2012/03/03: The first version of the H.264 codec is going to be released within the next month!

2012/03/15: The first version of the H.264 codec has been released!

The source code for the low level codec tools will appear and then be added to, periodically. We would like to offer these while the code is being developed to encourage comments early on in the design. Some of these raw classes may also be useful for other implementations.

We would like the community to contribute in conformance testing to the H.264 standard and faster implementations that use the defined interfaces.

Please do not use H.264 decoder from this package, it does not work. Microsoft's or LAV decoder can decode H.264 without any problem:



Please note that the use of the H.264 video standard may require the payment of license fees. These are governed by the MPEG Licensing Authority and the rules may be found at www.mpegla.com. It is your responsibility, not the authors of this source code, to ensure that you comply.

Developers

Join this project:

To join this project, please contact the project administrators of this project, as shown on the [project summary page](#).

Get the source code:

Source code for this project may be available as [downloads](#) or through the CVS or Subversion SCM repository used by the project, as accessible from the [project summary page](#).

