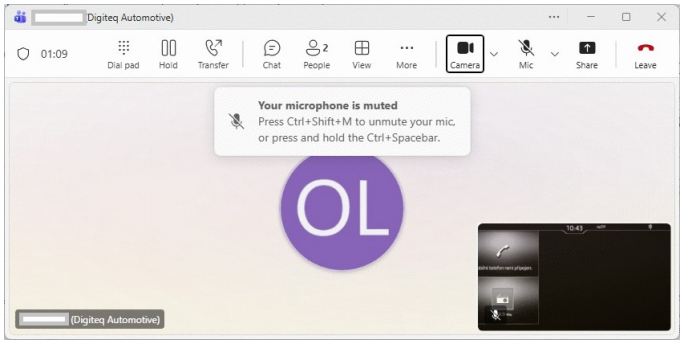
FG4 DirectShow CookBook

Please note that FG4 Frame grabber implements generic video source and video renderer. It means that even applications not designed for FG4, can work with it without problems. E.g. you can display FG4 video in Microsoft Teams, VLC, OBS Studio and many others.

This document is intended for FG4 users, but anybody who is working with DirectShow video could get inspiration here.

Special application like DQ Viewer is not necessary.



The drawback of this approach is that standardized API is more complex in comparison with custom library.

User can develop video processing filter without FG4 because of a standard API used. And after full debugging the filter will work with FG4.

CookBook is not intended to be a manual. It is only demonstration of chosen use cases.

Table of Contents

1 Direct Show concept..... 2
2 Direct Show components..... 2
3 Device configuration..... 3
4 Recommended configurations..... 4
4.1 Simple rendering..... 4
4.2 Rendering YUYV stream..... 6
4.3 Rotated view..... 6
4.4 Playing a video on display..... 7
4.5 Rendering a video to the file..... 7

4.6 Feedback....9
4.7 Feedback with preview....10
4.8 Mixing videos....11
4.9 Timestamp....12
4.10 Testing video contents....12
4.11 Multiple video consumers....13
4.12 Splitting video to 2 sub videos....14
5 Network....15
5.1 Connecting to multibox through raw TCP stream....15
5.2 Broadcasting through RTSP protocol....15
6 Show windows desktop on display....18
7 Starting tests automatically....18
8 Errata....19
8.1 Critical....19
Direct connection of Video generator and renderer causes hang during graph stops....19
8.2 Unpleasant....20
Microsoft's renderer to window does not work well....20
8.3 Minor....21
Microsoft's *AVI/WAV File Source* filter does not support flawless video framerate....21
Change on frame could get back-propagated....21
9 Direct Show retired?....22
10 Direct Show Filter list....22
10.1 CSIR VPP....26
10.2 DQ VPP....26
10.3 MediaLooks....26
10.4 VisioForge....27

1 Direct Show concept

A main concept of DirectShow is that any application that needs to support multimedia needs to run a graph.

Graph could be created in GUI application or directly in C++.

2 Direct Show components

Assembling Direct Show graph

There exists *GraphEdt.exe* application from Microsoft that could be found elsewhere

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\graphedt.exe

C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\x64\graphedt.exe

We recommend to use freeware *GraphStudioNext*

https://www.videohelp.com/software/GraphStudio

GraphStudioNext is storing filter properties into GRFX format and you open a graph with identical settings. If you use Microsoft's graphedit, you can be confronted with unpleasant surprise, when some filter parameter has different property value.

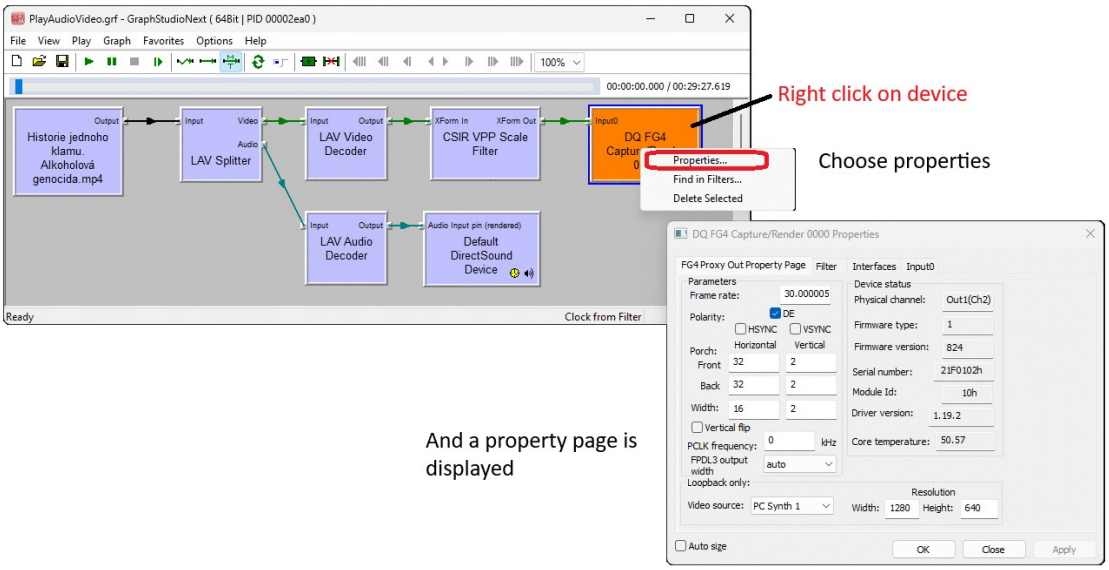
Please note that there are 2 variants 32bit and 64bit. On 64 bit system you can use both of them. But on 32 bit system you can use only 32 bit variant.

Microsoft standard renderers that are bundled in Windows:

https://learn.microsoft.com/en-us/windows/win32/directshow/about-video-rendering-in-directshow

3 Device configuration

All necessary options could be set through driver's property page. No special application like DQviewer needed.

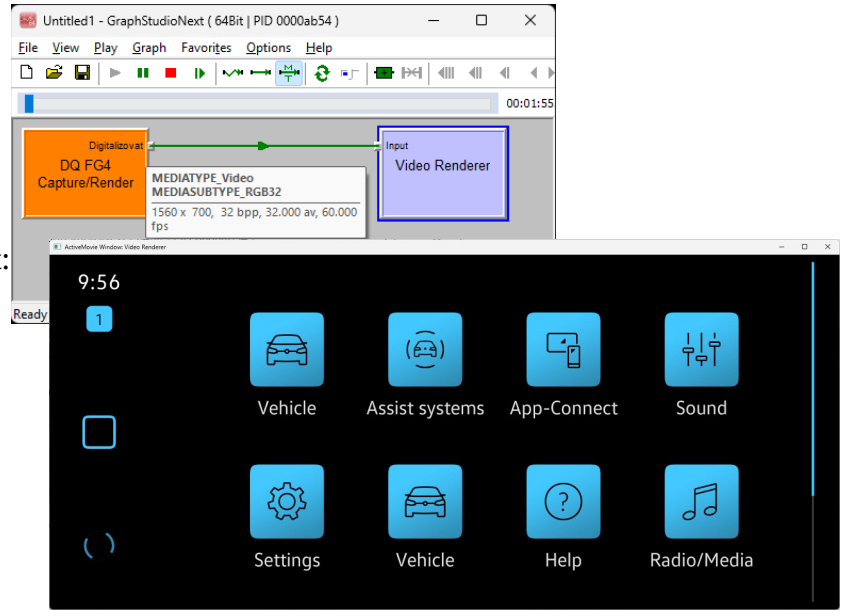


4 Recommended configurations

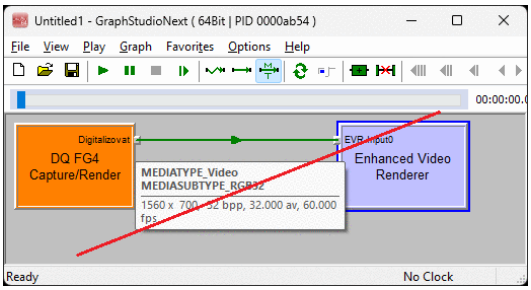
4.1 Simple rendering

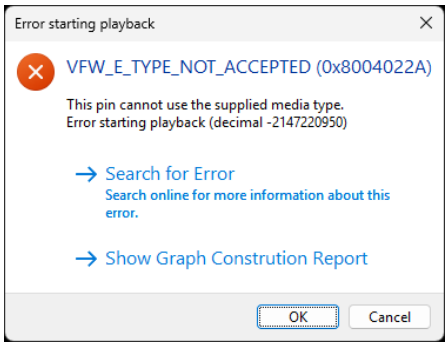
Insert block FG4 signal source and a standard Video renderer:

It should produce following result:

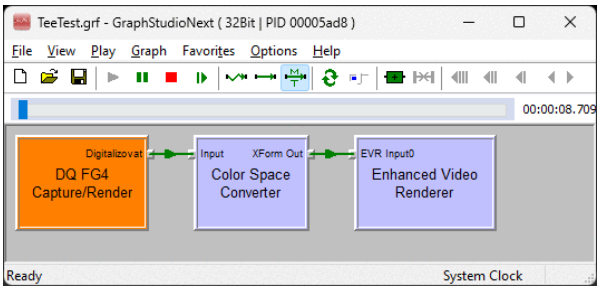


There is one pitfall. Enhanced Microsoft's renderer sometimes does not connect to a lower filter:

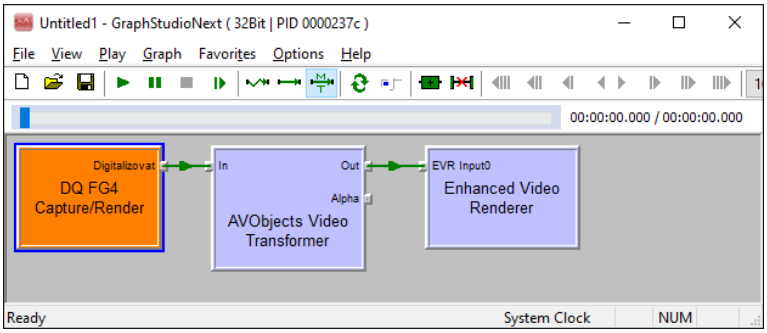


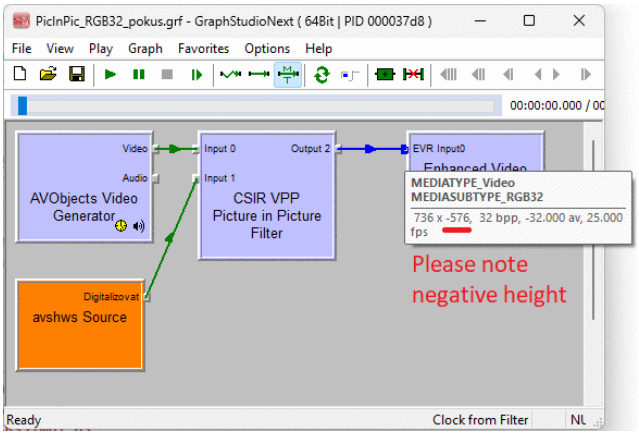


When this occurs, use middle **colorspace converter filter**, and everything works:



Or you can use free AV transform filter that provides a same job.
http://www.avobjects.com/products/freeware_filters/video_transformer.html





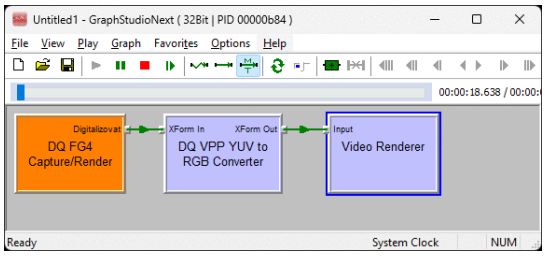
The problem behind is that Microsoft's renderer sometimes forces to use negative height or asks for different geometry, that lower driver/filter does not provide.

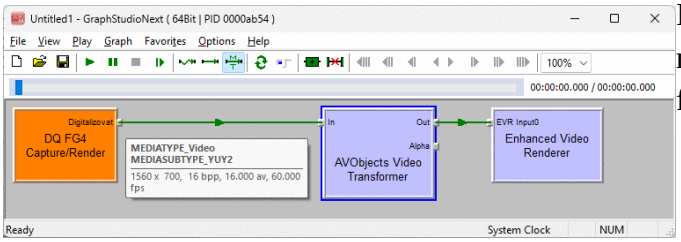
4.2 Rendering YUYV stream

FG4 device can optionally output YUYV stream. No Microsoft's renderer can directly render YUYV.

Standard Microsoft's colorspace converter is not capable of accepting YUYV format, see documentation here:
https://learn.microsoft.com/en-us/windows/win32/directshow/color-space-converter-filter

The conversion filter before rendering must be provided. We recommend DQ VPP YUV to RGB Converter.



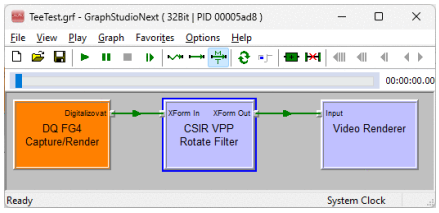


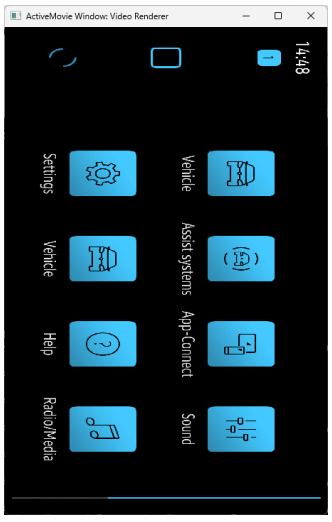
It is also possible to use free filter from AV objects named **Video transformer**. It is downloadable from here: http://www.avobjects.com/download

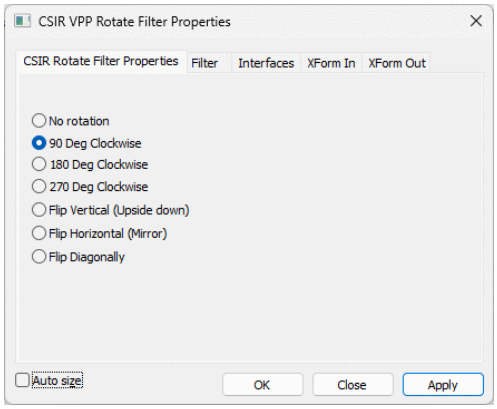
4.3 Rotated view

It is available free rotation DQ filter, installation is downloadable from here:
https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/

There also exists commercial **VideoRotator from AV Objects**: http://www.avobjects.com/download





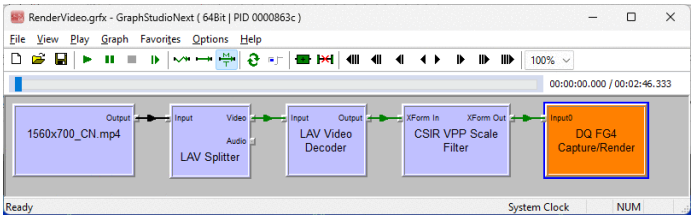


4.4 Playing a video on display

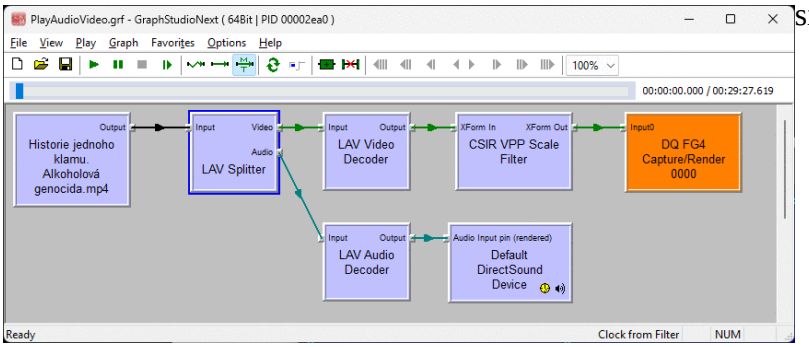
Rendering a videostream is a more complex task. Videostream usually does not have suitable resolution, so resize filter must be used. We recommend our modified DQ VPP scale filter. It is accessible from here:
https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/

Do not use original CSIR filters https://sourceforge.net/projects/videoprocessing/ **, they do not support RGB32 format needed by FG4. They are also critically buggy.**

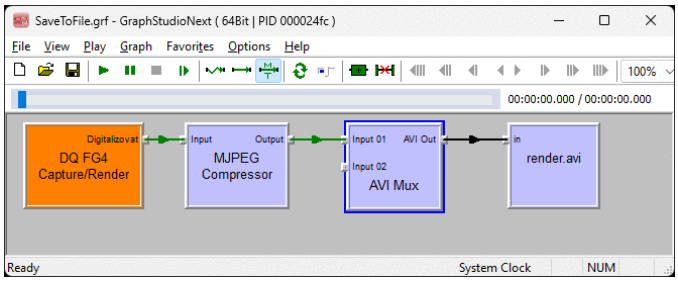
First filter **File Source (async)** is a part of Microsoft's Windows DShow. But the video stream must be decoded. We recommend to use LAV filters accessible from here:
https://www.videohelp.com/software/LAV-Filters



Just for a case when you want to play audio and video simultaneously:

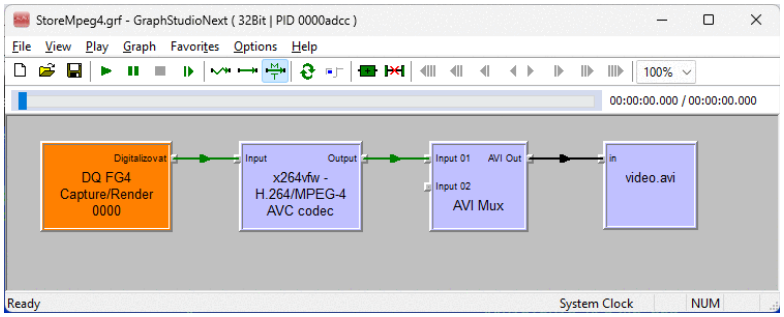


4.5 Rendering a video to the file

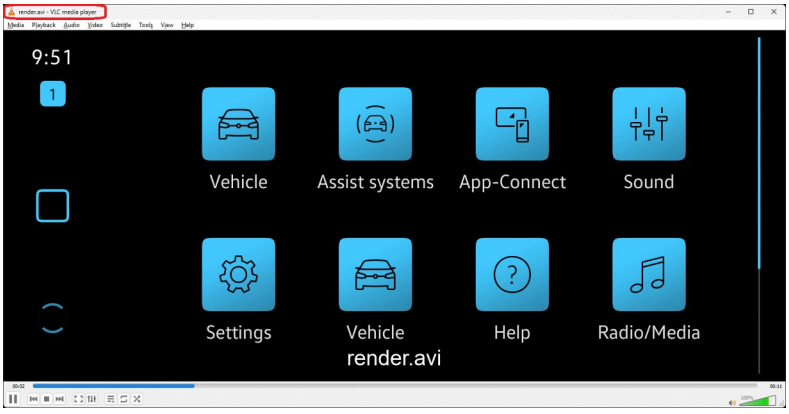


File writer needs some effective storage format. Microsoft's MJPEG is used for demonstration. It is possible to use commercial filter with H264 format:
http://www.avobjects.com/products/encoders/h_264_encoder.html
http://www.avobjects.com/products/encoders/qs_encoder.html
Free H264 codec also exists: https://sourceforge.net/projects/x264vfw/
https://hax264.sourceforge.net/

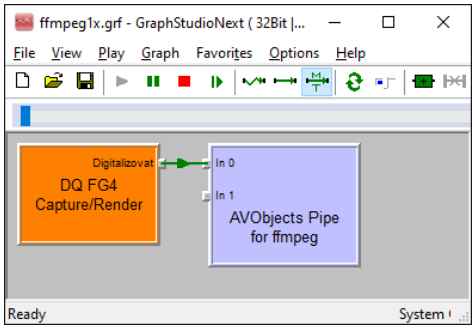
Generate H264 MPEG-4:

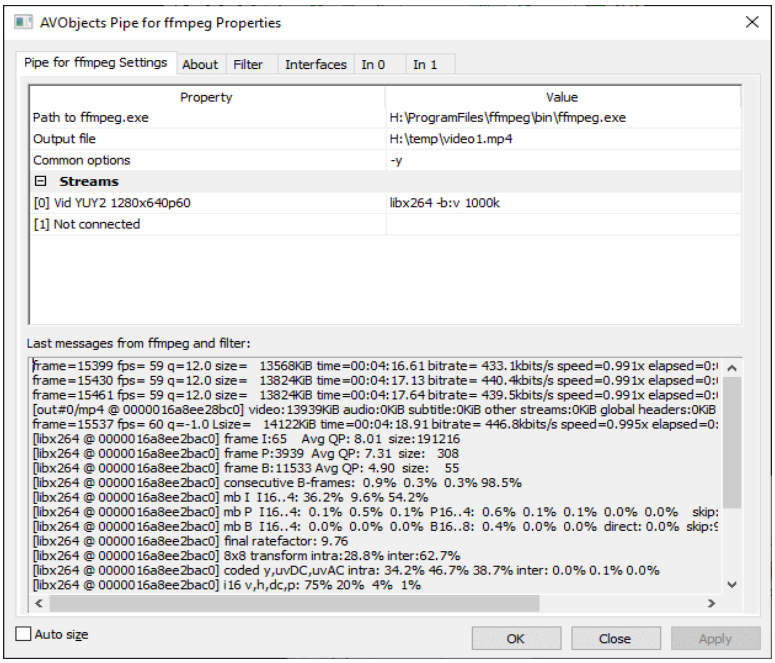


The file recorded could be played using VLC media player



Another possibility is to use "AV Objects Pipe for ffmpeg" as depicted on following images.



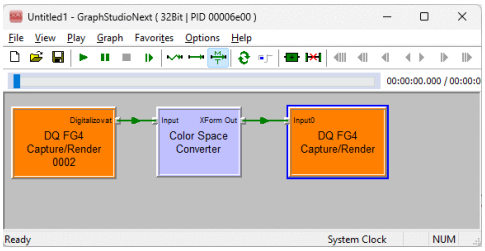


This DirectShow filter wraps ffmpeg. You will need to fill encoder manually (mpeg2video, mpeg4, libxvid, libx264, libx265, etc.) like "*libx264 -b:v 1000k*". File type could be mp4, mov, avi, mkv, mpg or else. Additional parameter should be ("-y").

Filter could be downloaded from here:
http://www.avobjects.com/products/special_filters/Pipe_for_ffmpeg.html
ffmpeg codecs are described here: https://ffmpeg.org/ffmpeg-codecs.html

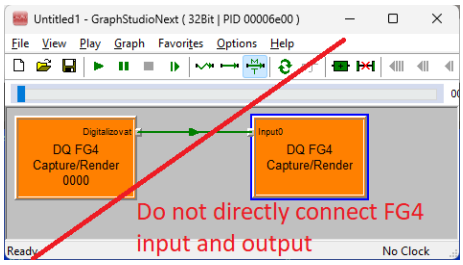
4.6 Feedback

This might be useful when you want to feed video from one FG4 card input to the output on different physical FG4 card. It is recommended to use internal FG4 loopback feature for video on one card.



There is some flaw in DirectShow and 2 kernel based filters should not be interconnected this way. It randomly crashes during video stop.

If you implement this connection in C++, you should stop video source first and then renderer. It works.

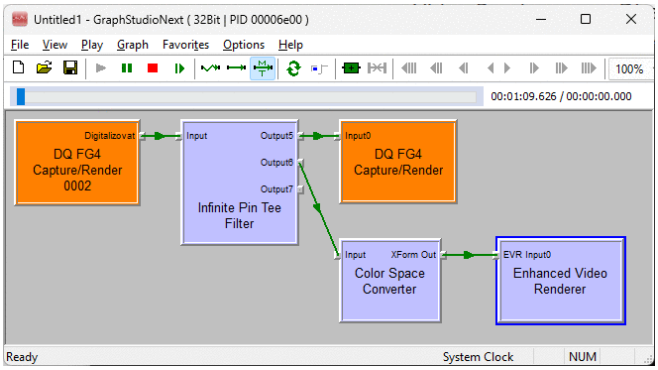


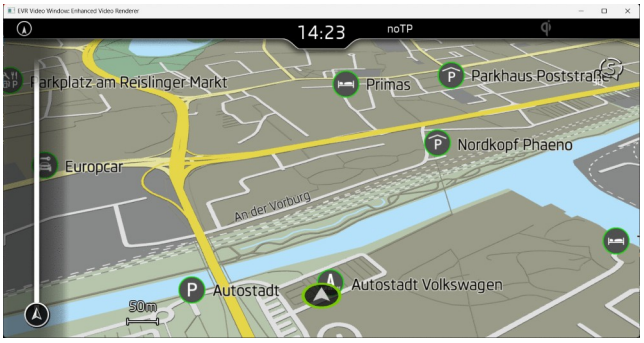
4.7 Feedback with preview

This grants that video grabbed from FG4 is displayed on screen and simultaneously sent to display.

Please note that Infinite Tee pin filter is a standard part of DirectShow and it is preinstalled on Windows. https://learn.microsoft.com/en-us/windows/win32/directshow/infinite-pin-tee-filter

No external component is needed for this setup.

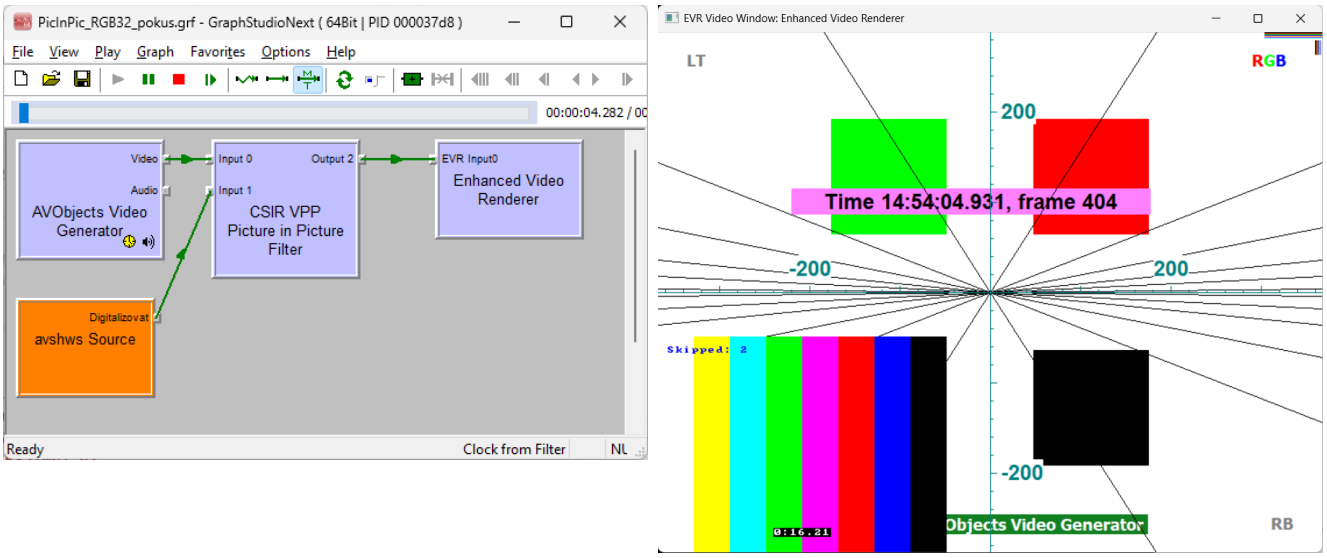




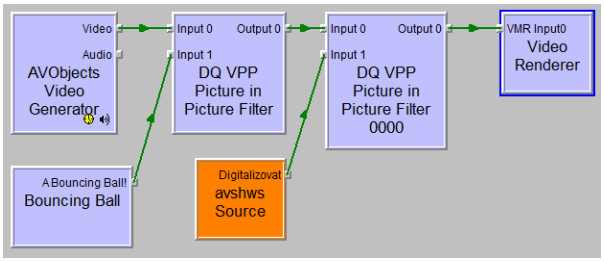
From unknown reason the direct connection is sometimes refused, thus Enhanced video renderer is connected through Color Space Converter as a workaround.

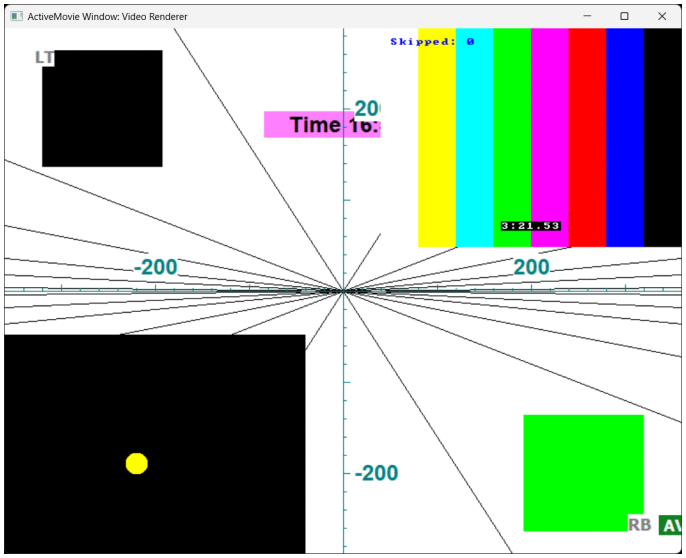
4.8 Mixing videos

It is possible to mix video from multiple sources, see image. This could be used for outputting something on display.



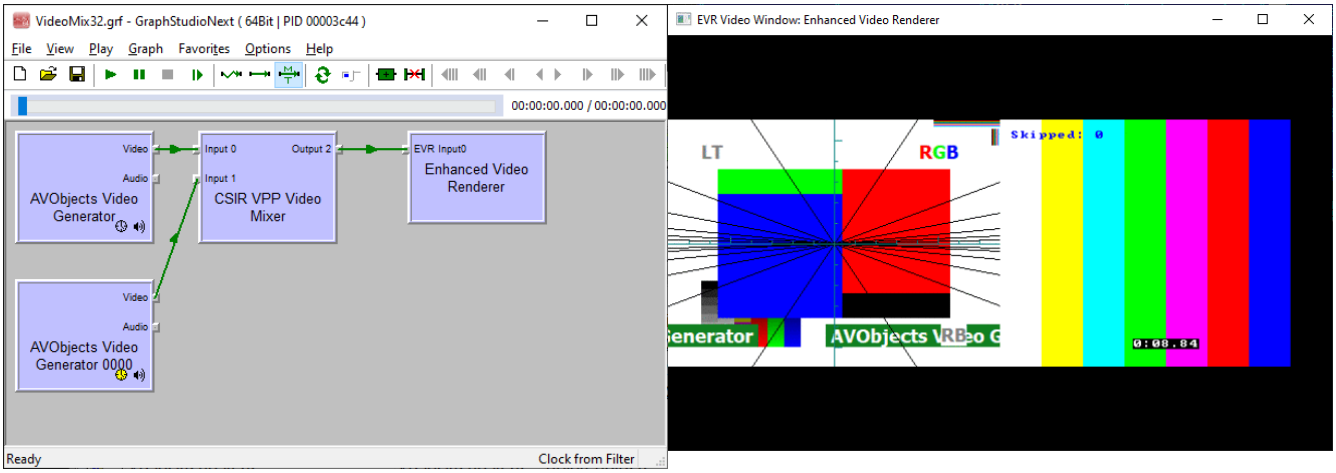
Just for a case when you need to insert two or more external video timestamps.



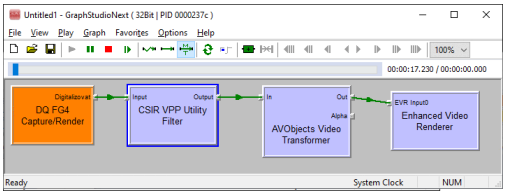


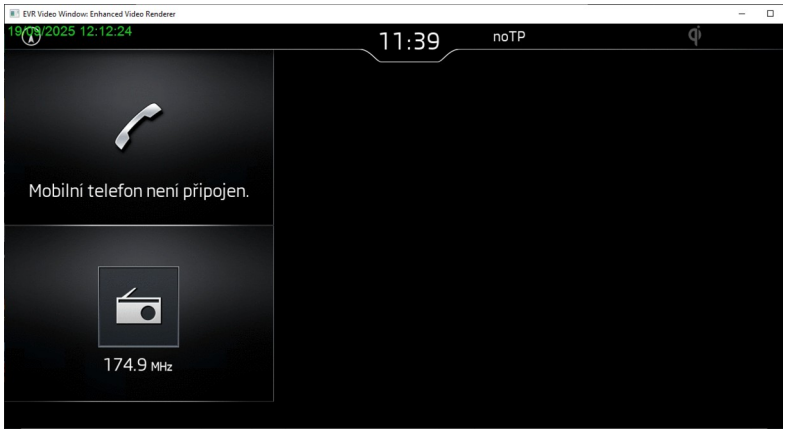
Please keep in mind that component should be simple enough. Two chained PicInPic filters are working with practically same CPU load as one mammoth all-in-one component. You should uncheck property that requests frame duplication at least in one PicInPic chained filter.

Another possibility is to mix 2 videos horizontally or vertically.



4.9 Timestamp

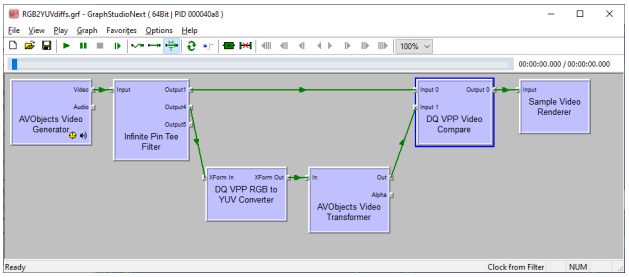


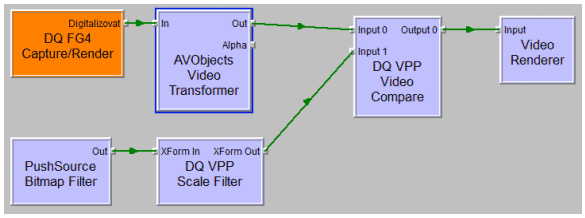


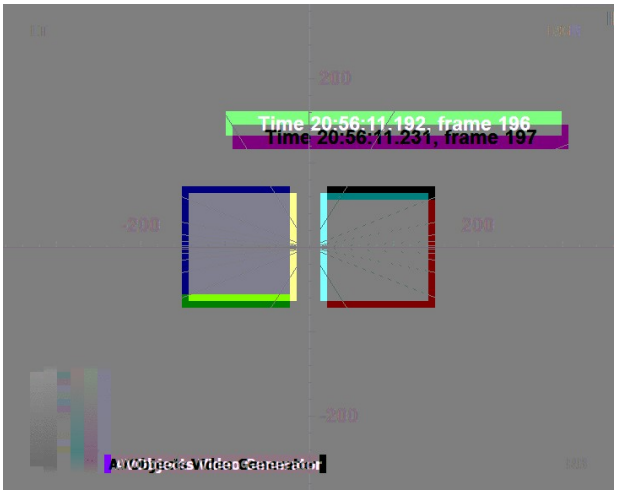
It is possible to insert timestamp or any debug information to the video. Please note a green text on a following image.

4.10 Testing video contents

We can have a task to compare 2 video streams. One videostream might contain solid picture and second gathered video. The situation is depicted on following 2 images.







You can use DQ VPP Compare filter and compare both videostreams. Result should be fully gray when frames have a same contents. Output video-stream could be further evaluated for summation of differences.

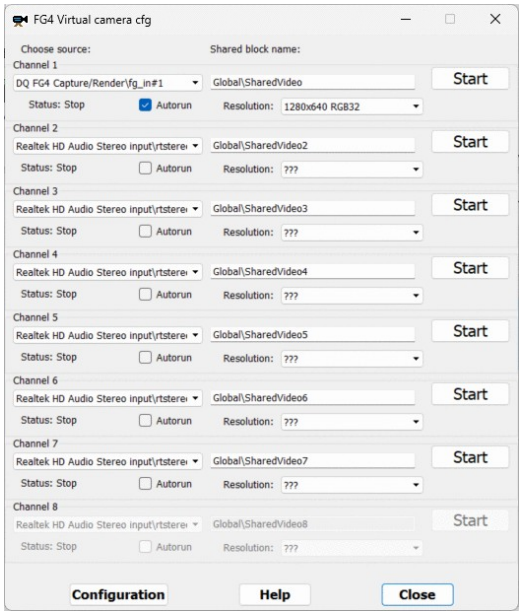
Converting from RGB to YUYV is allays lossy, it is here only for demonstration.

Possible use case might be to use *Push Source Bitmap filter.* This filter gets *sample.bmp* and sends it as video. You can compare bitmap contents to FG4 video. It is recommended to use proper resolution of source bitmap. Video scaling needs to rescale every frame and wastes CPU load.

It is also possible to compare 2 videosignals from 2 video grabbers.

4.11 Multiple video consumers

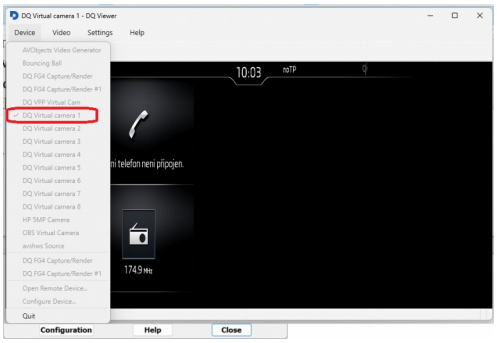
Camera kernel driver can feed only one camera source. But sometimes users need to connect to a camera from DQ Viewer, Python and some custom application simultaneously.

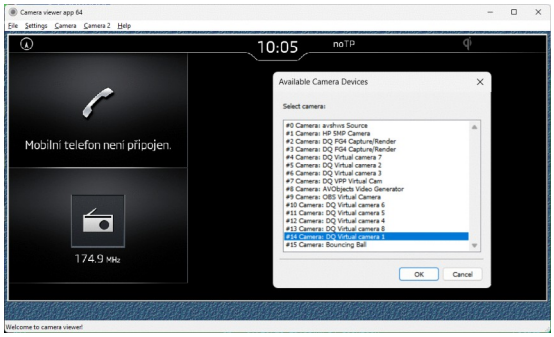


Please note that FG4 hardware is not capable to feed reliably multiple camera consumers. The DMA throughput is not sufficient enough to feed more video streams with granted maximal resolution and framerate.

Please note that virtual camera suffer these limitations:

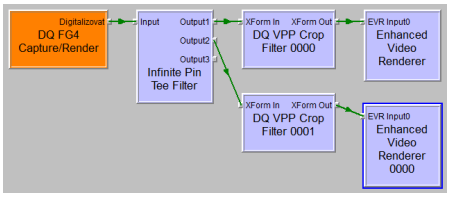
1) Does not grant frame synchronization
2) Does not prevent frame loss
3) Does not synchronize framerate
4) Problems with propagation resolution

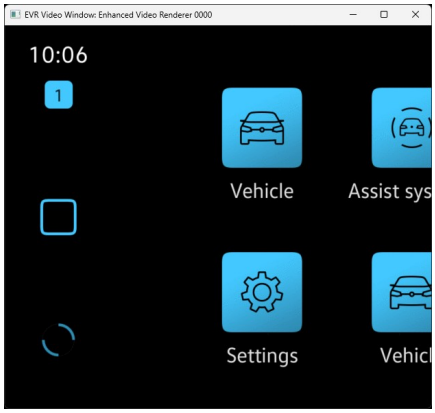


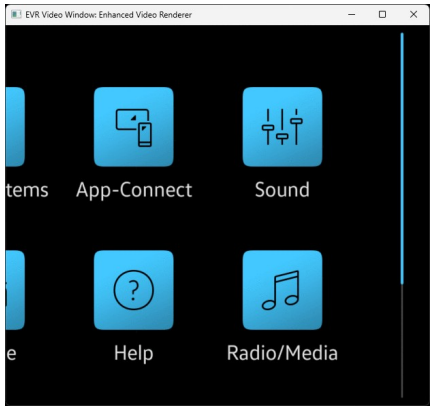


4.12 Splitting video to 2 sub videos

Sometimes there are sent 2 videosignals stacked in one frame. It is possible to create 2 (or even more) separate videosignals. It is recommended to use Microsoft's Infinite pin tee filter and DQ VPP Crop filters. Infinite tee pin filter does not copy image contents.





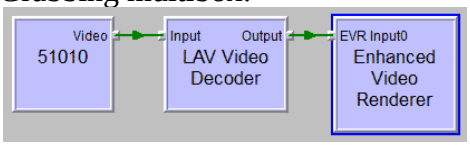


5 Network

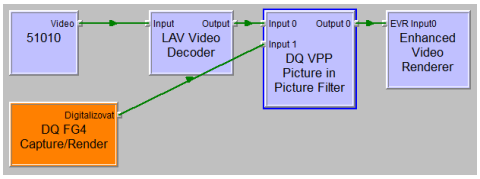
5.1 Connecting to multibox through raw TCP stream

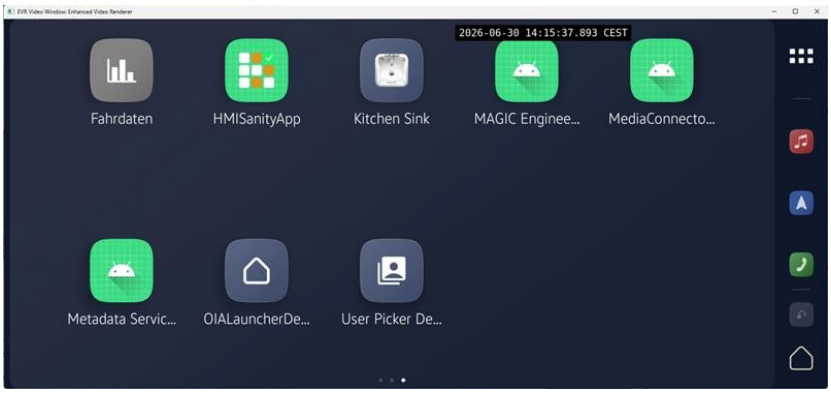
DQ multibox transmits videostream through TCP. *LAV Splitter Source filter* can directly open RAW TCP stream, like tcp://192.168.4.16:51010, and then you need to decode video.

Grabbing multibox:



Mix video from FG4 and Multibox:





It is possible to even mix a video from inserted FG4 negatively connected and video coming from Multibox. You can grab video on Multibox and transmit it from local FG4 or vice versa.

5.2 Broadcasting through RTSP protocol

I have found this way to be workable.

1, run *mediamtx* application without any parameter
https://sourceforge.net/projects/rtspsimpleserver.mirror/files/

2, List all dshow devices

```
ffmpeg -f dshow -list_devices true -i dummy
D:\Programs\ffmpeg-2026-01-29\bin>ffmpeg -f dshow -list_devices true -i dummy
ffmpeg version 2026-01-29-git-c898ddb8fe-essentials_build-www.gyan.dev Copyright
(c) 2000-2026 the FFmpeg developers
  built with gcc 15.2.0 (Rev8, Built by MSYS2 project)
  configuration: --enable-gpl --enable-version3 --enable-static --disable-
w32threads --disable-autodetect --enable-cairo --enable-fontconfig --enable-iconv
--enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-lzma --
enable-zlib --enable-libsrt --enable-libssh --enable-libzmq --enable-avisynth --
enable-sdl2 --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxvid --
enable-libaom --enable-libopenjpeg --enable-libvpx --enable-mediafoundation --
enable-libass --enable-libfreetype --enable-libfribidi --enable-libharfbuzz --
enable-libvidstab --enable-libvmaf --enable-libzimg --enable-amf --enable-cuda-llvm
--enable-cuvid --enable-dxva2 --enable-d3d11va --enable-d3d12va --enable-ffnvcodec
--enable-libvpl --enable-nvdec --enable-nvenc --enable-vaapi --enable-openal --
enable-libgme --enable-libopenmpt --enable-libopencore-amrwb --enable-libmp3lame --
```

```
enable-libtheora --enable-libvo-amrwbenc --enable-libgsm --enable-libopencore-amrnb
--enable-libopus --enable-libspeex --enable-libvorbis --enable-librubberband
  libavutil      60. 24.100 / 60. 24.100
  libavcodec     62. 23.103 / 62. 23.103
  libavformat    62.  8.102 / 62.  8.102
  libavdevice    62.  2.100 / 62.  2.100
  libavfilter    11. 11.101 / 11. 11.101
  libswscale      9.  3.100 /  9.  3.100
  libswresample   6.  2.100 /  6.  2.100
[dshow @ 00000236ee586bc0] "avshws Source" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_pnp_\\?
\root#unknown#0000#{65e8773d-8f56-11d0-a3b9-00a0c9223196}\global"
[dshow @ 00000236ee586bc0] "HP 5MP Camera" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_pnp_\\?
\usb#vid_30c9&pid_00c1&mi_00#6&e5cacbe&0&0000#{65e8773d-8f56-11d0-a3b9-
00a0c9223196}\global"
[dshow @ 00000236ee586bc0] "DQ FG4 Capture/Render" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_pnp_\\?
\pci#ven_1ed8&dev_0101&subsys_00000000&rev_00#8&29b27679&0&0008001800e8#{65e8773d-8
f56-11d0-a3b9-00a0c9223196}\fg_in#1"
[dshow @ 00000236ee586bc0] "DQ FG4 Capture/Render" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_pnp_\\?
\pci#ven_1ed8&dev_0101&subsys_00000000&rev_00#8&29b27679&0&0008001800e8#{65e8773d-8
f56-11d0-a3b9-00a0c9223196}\fg_in#2"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 7" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{2C29CC20-6A45-41A7-9688-97012BD4FDAF}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 2" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{5DB688F1-434B-4F38-A760-CD28502C1A9A}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 3" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{69746EE5-FE38-4495-8FEF-8C99091B8D56}"
[dshow @ 00000236ee586bc0] "AVObjects Video Generator" (audio, video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{966FCC90-1C7A-42C3-A2DF-F91F0BEADDA1}"
[dshow @ 00000236ee586bc0] "OBS Virtual Camera" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{A3FCE0F5-3493-419F-958A-ABA1250EC20B}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 6" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{BA0B8C0E-CA87-4585-A21F-16DD906F5FFE}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 5" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{C5B0A0FE-342E-4FF0-9185-5FF1EC2E71ED}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 4" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{E5F3FA29-CC18-4325-9731-7D905BFABE92}"
```

```
[dshow @ 00000236ee586bc0] "DQ Virtual camera 8" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{E98E8032-DA7A-42D6-AAC4-6784E8A9CAEC}"
[dshow @ 00000236ee586bc0] "DQ Virtual camera 1" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{FAC5A6DF-E083-4331-9228-B5D2787ADD7C}"
[dshow @ 00000236ee586bc0] "Bouncing Ball" (video)
[dshow @ 00000236ee586bc0]   Alternative name "@device_sw_{860BB310-5D01-11D0-
BD3B-00A0C911CE86}\{FD501041-8EBE-11CE-8183-00AA00577DA1}"
[dshow @ 00000236ee586bc0] "Pole mikrofonu (Technologie Intel® Smart Sound pro
digitální mikrofony)" (audio)
[dshow @ 00000236ee586bc0]   Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-
00A0C911CE86}\wave_{E5104487-ADF2-4D94-B349-81AFEA2FA9FD}"
Error opening input file dummy.
```

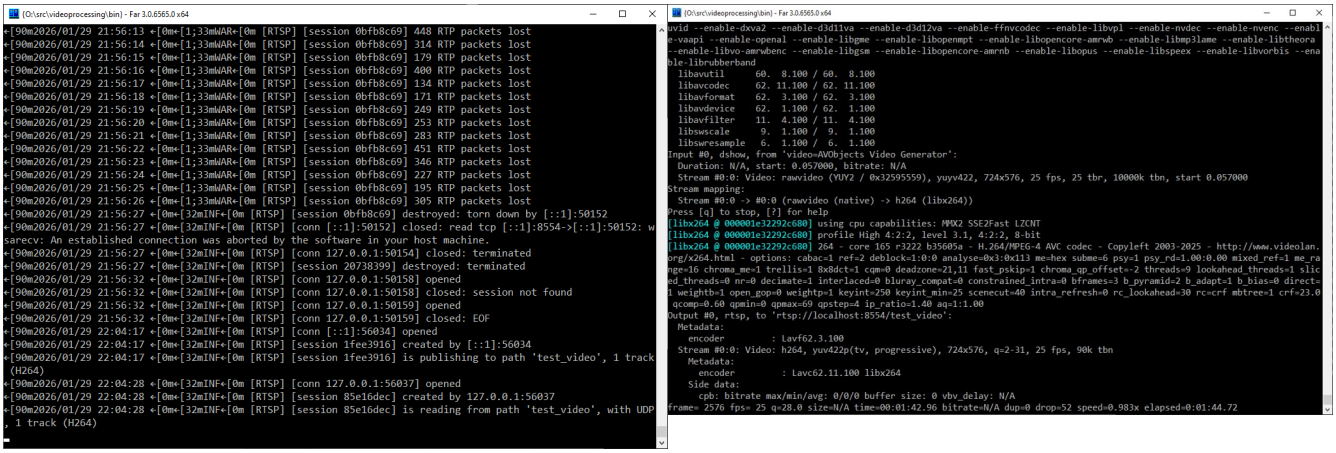
3, run *ffmpeg*

This works for DQ FG4 device:

ffmpeg -f dshow -i video="DQ FG4 Capture/Render" -c:v mpeg2video -preset fast -c:a aac -f rtsp rtsp://localhost:8554/test_video

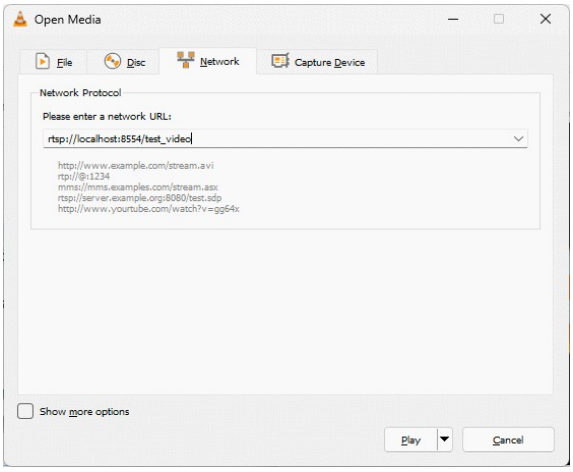
This command line could be used for sample video generartor:

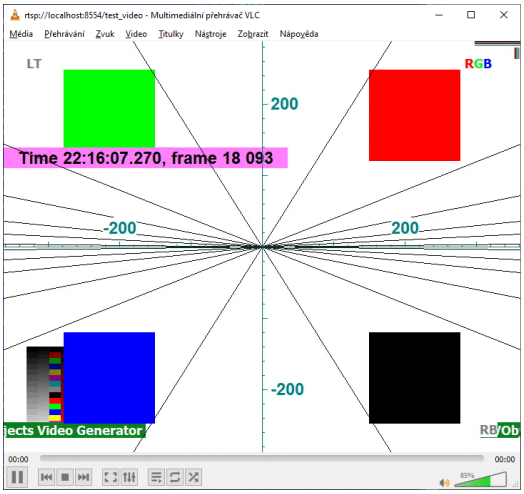
ffmpeg -f dshow -i video="avshws Source" -c:v libx264 -preset fast -c:a aac -f rtsp rtsp://localhost:8554/test_video



Then your RTSP server is up.

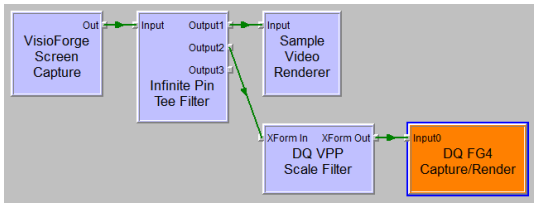
4, Now you can open video in VLC:



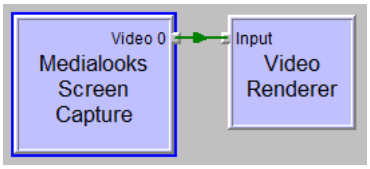


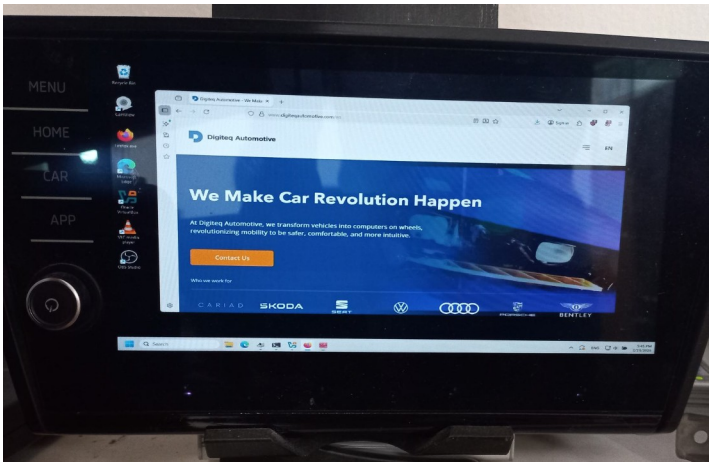
6 Show windows desktop on display

It is possible to grab desktop contents and send it to FG4 display.



MediaLooks's filter could be also used:





Screen capture from MediaLooks seems to work better. It can grab window from any application on desktop. On the other side it suffers with high CPU load.

7 Starting tests automatically

Graph Studio Next can run graph with command line.

"D:\Program Files\graphstudionext\graphstudionext.exe" C:
\JFO\PCIe_Video\baseclasses\videoprocessing\graphs\PicInPic\PiRGB24.grfx -run

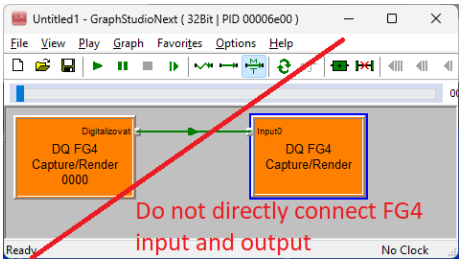
```
Command Line Options for the Program:

graphstudionext.exe [options] [file/moniker] [options]
-> file (optional)       -> a graph file or a video-file to render
-> options (optional)    -> one or more of the following switches:
    -a                       -> connect to a remote graph
    -run                     -> runs the graph if a file was rendered
    -exitafterrun            -> closes the application after the graph was stopped
    -noclock                 -> disables the clock if a file was rendered
    -progressview            -> opens the application in the progress view
    -exitonerror             -> closes the application on an error (no error dialog)
                                and sets the error code as the exit code
    -filters                 -> opens the filters dialog on startup
    -?                       -> opens this window on startup
```

8 Errata

Direct Show is Microsoft's product. It looks like Microsoft did not develop this well, DShow is pretty complex and fragile. There are a lot of problems in preinstalled components in Windows, that cannot be repaired and Microsoft will never fix them.

8.1 Critical

Direct connection of Video generator and renderer causes hang during graph stops



This configuration randomly crashes during video stop. The hang is unpleasant because user space Microsoft's component KS proxy creates deadlock against kernel code.

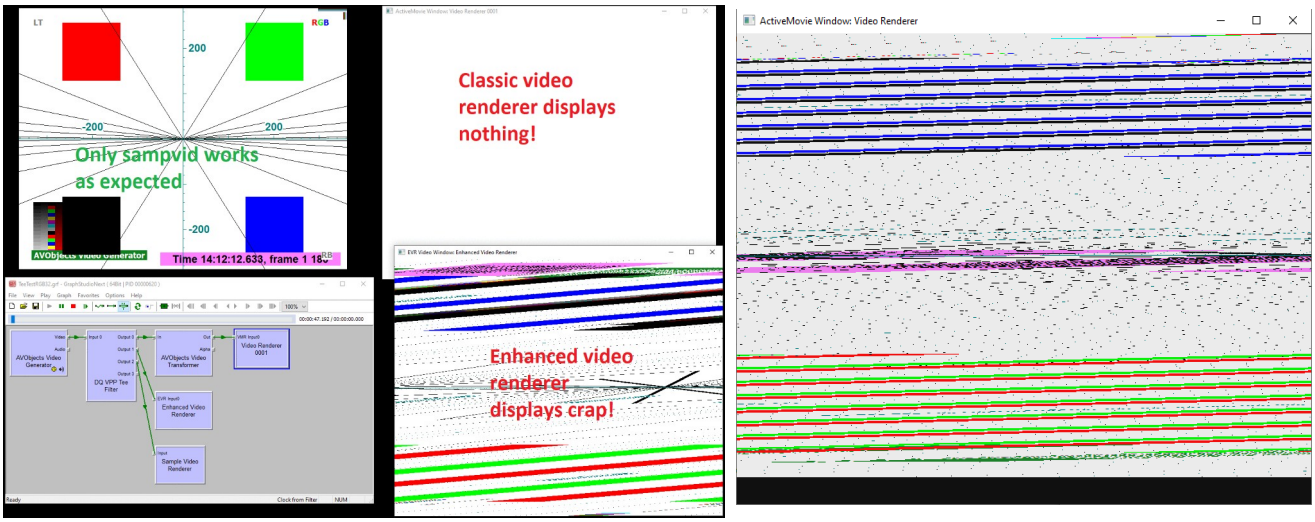
There are 2 possible workarounds:

1, Use intermediate filter
2, If you implement this connection in C++, you should stop video source first and then renderer. It works.

8.2 Unpleasant

Microsoft’s renderer to window does not work well

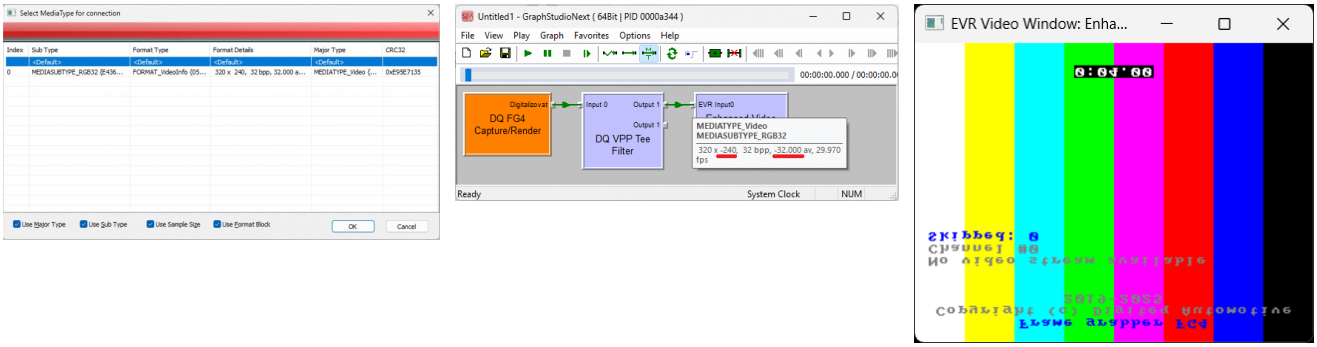
It sometimes displays video upside down, sometimes displays distorted video and sometimes displays only black or white contents. It looks like problem of another filters, but the failing component is usually Microsoft’s renderer. Microsoft expects that it could run faster when frame allocator from renderer is used. But in many cases this allocator cannot be used and intermediate filter is slower than internal frame data copy.



The proposed solution is to use “sampvid” renderer.

Microsoft’s renderer forces negative height

Microsoft’s renderer behaves really nasty. It does not accept negative height provided by video source. But in a final phase of video negotiation process it suddenly forces negative height.



This causes either break of graph or it is producing video upside down.

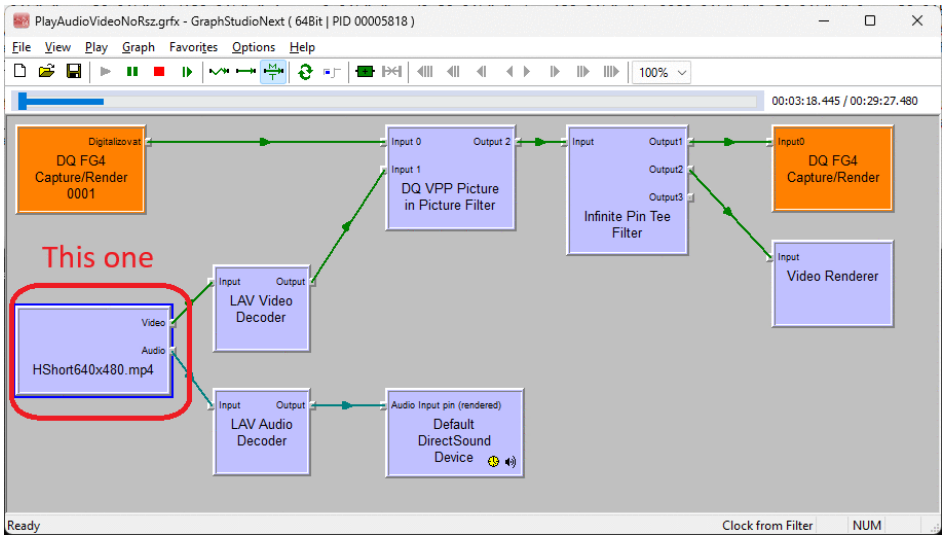
Most of DQ VPP filters have implemented fix for unexpected negative height video request. In this case they rather accepts negative height and provide internal video flip.

8.3 Minor

Microsoft's *AVI/WAV File Source* filter does not support flawless video framerate

It is generating lags.

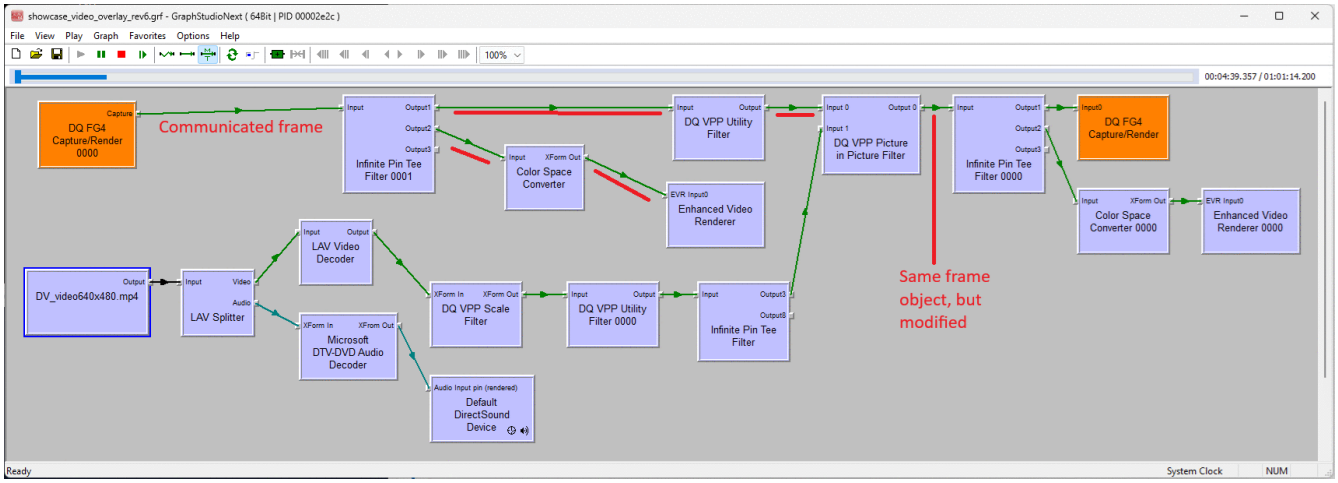
https://learn.microsoft.com/en-us/windows/win32/directshow/avi-wav-file-source



Prefer ***LAV Splitter Source filter*** instead.

Change on frame could get back-propagated

Some filters are patching frame data only without creating a new frame. It is much more faster than whole frame duplication. Typically Infinite Tee Pin filter is not duplicating video, but only broadcasting frames. In some case it could cause unwanted frame data gets to another branch.



9 Direct Show retired?

Microsoft claims that DirectShow is to be retired. It proposes a new API for camera.

It looks like using big fat library does a miracle.

But there are many problems:

1. New camera API has no support for rendering device (FG4 has 2 cameras and 2 renderers)
2. There are no available open source samples. There are no free tools that exists for Direct Show. There is no information how to write kernel based video generator.
3. New camera API is permanently changing
4. New camera API is available on Windows 11 only. Version used on Windows 10 is different and unsupported. Direct Show has 25 year history and many tools both commercial and free available.

10 Direct Show Filter list

There exists much more filters available. These ones has been chosen for demonstration what could be done with FG4 video.

Video Renderer	Microsoft / free	This renderer is good to render FG4 videostream. It has no extra feature requirements https://learn.microsoft.com/en-us/windows/win32/directshow/video-renderer-filter
Enhanced Video Renderers	Microsoft / free	This filter sometimes requires special DirectShow features https://learn.microsoft.com/en-us/windows/win32/directshow/enhanced-video-renderer-filter
Color Space Converter filter	Microsoft / free	Sometimes this filter is necessary to modify videostream https://learn.microsoft.com/en-us/windows/win32/directshow/color-space-converter-filter
Infinite Pin Tee Filter	Microsoft / free	This filter allows to split video to 2 or more identical videos. https://learn.microsoft.com/en-us/windows/win32/directshow/infinite-pin-tee-filter
AVI Mux Filter	Microsoft / free	Helper for generating AVI videostream https://learn.microsoft.com/en-us/windows/win32/directshow/avi-mux-filter
File Source (Async) Filter	Microsoft / free	The Async File Source filter opens and reads local files of many different data formats and passes the data to a parser filter. https://learn.microsoft.com/en-us/windows/win32/directshow/file-source--async--filter
File Writer Filter	Microsoft / free	The File Writer filter can be used to write files to disc regardless of format. The filter simply writes to disc whatever it receives on its input pin, so it must be connected upstream to a multiplexer that can format the file correctly. https://learn.microsoft.com/en-us/windows/win32/directshow/file-writer-filter
MJPEG Decompressor Filter	Microsoft / free	This filter decodes a video stream from motion JPEG to uncompressed video. Some digital video cameras produce a motion JPEG video stream. https://learn.microsoft.com/en-us/windows/win32/directshow/mjpeg-decompressor-filter
Ball	Microsoft SDK compilled J.Fojtik / free	Very simple video source filter that is intended for programmers to learn how to write these filters. Original location: https://github.com/microsoft/Windows-classic-samples/tree/main/Samples/Win7Samples/multimedia/directshow/filters/ball Precompiled: https://sourceforge.net/projects/direct-show-demo-filters/files/SampleFilters/
AVSHWS	Microsoft DDK compilled J.Fojtik/ free	Very simple Windows kernel video source filter that is intended for programmers to learn how to write these filters. http://ftsoft.com.cz/CamView/AVSHWS/index.htm

EZRGB24	Microsoft SDK	This filter creates fancy effects in video: https://github.com/microsoft/Windows-classic-samples/tree/main/Samples/Win7Samples/multimedia/directshow/filters/ezrgb24
	improved J.Fojtik/ free	Precompiled and improved: https://sourceforge.net/projects/direct-show-demo-filters/files/SampleFilters/ Improved EZRGB24 filter accepts also RGB32 produced from FG4.
Sampvid	Microsoft SDK	This filter demonstrates, how to display video into MS Window. https://github.com/microsoft/Windows-classic-samples/tree/main/Samples/Win7Samples/multimedia/directshow/filters/sampvid
	improved J.Fojtik/ free	Precompiled and improved: https://sourceforge.net/projects/direct-show-demo-filters/files/SampleFilters/
PushSource	Microsoft SDK	This filter allows to read BMP file and send its contents as video. https://learn.microsoft.com/en-us/windows/win32/directshow/push-source-filters-sample
AV Video Generator	AVObject / free	DirectShow filter for generating a video stream with the necessary parameters, for creating video frames for checking chroma key, for outputting an audio stream synchronized with video, and much more. http://www.avobjects.com/products/freeware_filters/video_generator.html
AV Object pipe for ffmpeg	AVObject / commercial	Pipe for ffmpeg uses the *ffmpeg.exe* external module to encode video and audio streams and write them to files. The encoding and file types depend on the ffmpeg.exe used. http://www.avobjects.com/products/special_filters/Pipe_for_ffmpeg.html
AV QS encoder	AVObject / commercial	DirectShow filter for encoding 8-bit 4:2:0 progressive or interlaced video frames in HEVC, H264 or MPEG-2 formats. http://www.avobjects.com/products/encoders/qs_encoder.html
AV Video Transformer	AVObject / free	DirectShow filter for conversions of uncompressed video stream. http://www.avobjects.com/products/freeware_filters/video_transformer.html
Spy Filter	AVObject / free	DirectShow filter for tracing (together with the log object) connections, positioning, data flow, etc. http://www.avobjects.com/products/freeware_filters/Spy_filter.html
X264vfw	Masternobody / free	Compression video to H246 stream https://sourceforge.net/projects/x264vfw/
LAV Splitter	LAV / free	Splits audio and video from AVI https://www.videohelp.com/software/LAV-Filters
LAV Video	LAV / free	Decodes compressed videoformat from video file.

Decoder		https://www.videohelp.com/software/LAV-Filters
LAV Splitter Source filter	LAV / free	Reads input file and splits audio and video from AVI; can read also raw TCP https://www.videohelp.com/software/LAV-Filters
DQ/CSIR Scale Filter	CSIR reworked in DQ / free	Allows to scale video resolution. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
DQ/CSIR Rotate filter	CSIR reworked in DQ / free	Allows to rotate video. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
DQ/CSIR Picture in Picture Filter	CSIR reworked in DQ / free	Allows to insert 2nd video video as thumb. https://sourceforge.net/projects/direct-show-demo-filters/files/CSIR_VPP_VIDEOPROCESSING/
DQ/CSIR Utility Filter	CSIR reworked in DQ / free	Allows to add watermark text to video. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
DQ/CSIR Crop Filter	CSIR reworked in DQ / free	Allows to crop video. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
DQ/CSIR Video Mixing Filter	CSIR reworked in DQ / free	Allows to mix 2 videos horizontally or vertically. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
DQ YUV to RGB Converter	DQ / free	Converts many YUV based formats to RGB24 or RGB32. https://sourceforge.net/projects/direct-show-demo-filters/files/DQ_VPP_VIDEOPROCESSING/
Visio Forge screen capture	Visio Forge/ commercial	Visio Forge DShow suite https://www.visioforge.com/processing-filters-pack
Screen Capture DirectShow Filter	MediaLooks/ shareware commercial	Fancy filter for converting screen contents into DShow video https://www.free-codecs.com/download/medialooks_screen_capture_directshow_filter.htm

10.1 CSIR VPP

Original CSIR filters are located here https://sourceforge.net/projects/videoprocessing/ **, they mostly do not support RGB32 format needed by FG4.** They do not support persistence, that allows to save filter state from any 3rd party software. They have a lot of bugs that are causing to crash.

It seems that CSIR VPP development ceased and project is currently death.

10.2 DQ VPP

This set of filters has been derived from CSIR VPP. There has been a lot of improvements implemented. This filter set is intended to provide some filters for experiments with video produced by FG4.

Most significant improvements are:

- Support of RGB32 format needed by FG4
- More other videoformats supported
- Many bugfixes
- Added unittests for some operations
- Filters could be directly connected to Microsoft's renderer
- Filters run significantly faster

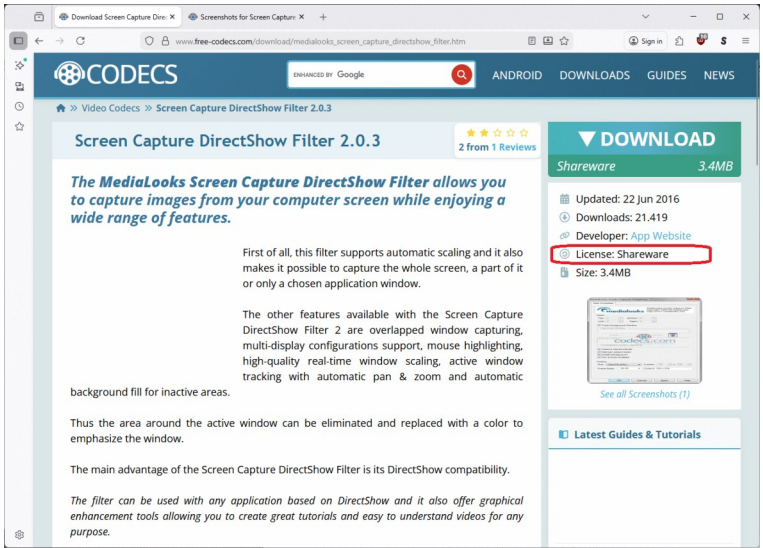
Samples of speedup DQ_VPP over CSIR_VPP tested on Inter core i7:

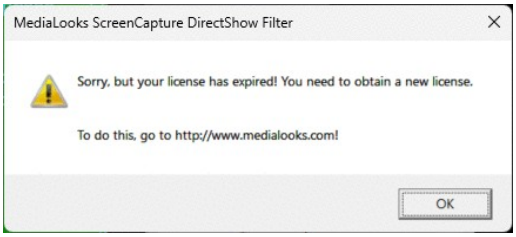
Conversion from RGB32 to YUV 1920×1080	fix point 2.209[ms]	orig 6.485[ms]	**3x**
Conversion AYUV to RGB24 1920×1080	time 2.767[ms]	orig 18.347[ms]	**6.6x**
Scale YUV420 from 1920×1080→800×600	time 11.613[ms]	orig 27.444[ms]	**2.4x**
Scale RGB32 from 1920×1080→800×600	SSE 8.109[ms]	orig 22.025[ms]	**2.7x**

The problem behind is that DQ_VPP uses doubles for pixel processing. It is ineffective and slow. This improvement might be sufficient for 2k video. 2048×1556 on 30fps needs to process 96MPix/s. But 4k video would need processing, that uses GPU.

10.3 MediaLooks

Screen capture from MediaLooks is declared to be a shareware. License is not fully clean. It looks that this bundled product can be freely used without payment and only SDK is payed as described here: https://medialooks.com/video-sdk#rec825278449





This component claimed to be free for evaluation as shareware. **It is not true – component is time limited demo version!**
For commercial use, I have obtained following response from MedialLooks:
To clarify, we stopped working with the Screen Capture DirectShow filter about 10 years ago. This product is now deprecated but can be obtained "as is" for $750. We have replaced it with our specialized SDKs for our customers to provide a more robust and integrated solution.

10.4 VisioForge

VisioForge provides several DirectShow filters https://www.visioforge.com/processing-filters-pack
Cost of Processing Filters pack including screen capture filter is EUR 350.

They promise this functionality

Video Processing and Effects Filter

- Image overlay (BMP, JPEG, PNG, GIF, RGB data)
- Text overlay (choose font, style, size, etc.)
- Video transparency
- Brightness, contrast, saturation, hue adjustments
- Deinterlace (several algorithms available based on quality and performance)
- Denoise
- Motion detection (with motion level detection, matrix, and highlights)
- Chroma-key (GPU-optimized)

Additional Filters

- Video resize filter
- Video crop filter
- Video flip/rotate filter
- PIP (Picture-In-Picture) filter
- Screen capture filter (Direct3D 9/Direct3D 11/DXGI/Desktop Duplication)
- Video color space converters (RGB and YUV formats)
- DirectX - accelerated video processing

There is one big potential problem that these filters need C++ API to be fully configured. No well known common API. It might be useful for some usecases.

If you really need to use C++ we are publishing FG4 SDK - FG4_SDK_ver1.zip here https://fg4.digiteqautomotive.com/ This SDK demonstrates how to pick FG4 video source from C++. It is not easy and it needs knowledge of DirectShow/COM/C++.