



VIDEO COMPRESSION GURU

Elecard DirectShow Codec SDK v.4.1

Reference Manual

Notices

Elecard Components Reference Manual

First edition: April, 2008.

Date modified: 9 November, 2018.

For information, contact Elecard.

Tel: +7-3822-488-585.

More information can be found at: <http://www.elecard.com>

For Technical Support, please contact the Elecard Technical Support Team: tsup@elecard.com

Elecard provides this publication “as is” without warranty of any kind, either expressed or implied.

This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Elecard may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Other company, product, trademarks, and service names are trademarks or service marks of other companies or corporations.

Copyright © 2008-2018 Elecard. All rights reserved.

CONTENTS

1. INTRODUCTION	4
1.1 ABOUT THIS DOCUMENT	4
1.1.1 Purpose	4
1.1.2 Topics Covered	4
1.1.3 Related Documentation	4
1.2 PREFACE	4
1.2.1 Documentation	4
1.2.2 Components	5
1.2.3 DirectShow Base Classes	7
1.2.4 Elecard Base Classes	7
1.2.5 Sample Applications	7
1.3 REQUIREMENTS	8
1.3.1 Hardware Requirements	8
1.3.2 Software Requirements	9
1.4 TECHNICAL SUPPORT	9
2. GETTING STARTED	10
2.1 INTRODUCTION	10
2.2 INSTALLING ELECARD DIRECTSHOW CODEC SDK	10
2.3 UNINSTALLATION ELECARD DIRECTSHOW CODEC SDK	10
2.4 DESCRIBING CODEC SDK FOLDER STRUCTURE	10
2.5 DIRECTSHOW FILTERS ACTIVATION	11
2.5.1 Instance Activation	11
2.5.2 Adding a DirectShow Filter to the Graph without Registration	12
3. DIRECTSHOW OVERVIEW	13
3.1 INTRODUCTION	13
3.2 DIRECTSHOW MAIN TERMS AND DEFINITIONS	13
3.2.1 DirectShow Filter Types	13
3.2.2 Data Flow Principle	13
3.3 DIRECTSHOW UTILITY APPLICATIONS	14
3.3.1 Building a simple file playback graph using Elecard components	14
4. SAMPLE APPLICATIONS	17
4.1 INTRODUCTION	17
4.1.1 Building sample applications	17
4.2 ELECARD DIRECTSHOW CODEC SDK SAMPLE APPLICATIONS	18
4.2.1 Simple Player	18
4.2.2 Simple Player QS	19
4.2.3 Console Player	20
4.2.4 Simple Encoder	21
4.2.5 Simple Encoder QS	23
4.2.6 Console Encoder	24
4.2.7 Simple Capture	25
4.2.8 Low Delay Streamer	26
4.2.9 Low Delay Client	28
4.2.10 NWPlayer	30
4.2.11 NWServer	35
4.2.12 NWServerTranscoder	37

1. Introduction

1.1 About This Document

1.1.1 Purpose

This document provides an overview of the installation, set up and use of the Elecard DirectShow Codec SDK. It includes information about the structure of the Elecard DirectShow Codec SDK, provides a quick overview of the DirectShow fundamentals and features and a detailed description of the components and sample applications.

1.1.2 Topics Covered

The following lists the topics covered in this document:

- **Section 1: Introduction** – provides a general overview of the SDK and describes the purpose of the document.
- **Section 2: Getting Started** – describes how to install, uninstall, and run the SDK. This section also provides information on the Elecard DirectShow Codec SDK folder structure.
- **Section 3: DirectShow Overview** – provides a brief description of the basic concepts of the DirectShow system, including component model, filters, graphs, and more.
- **Section 4: Building Simple Filter Graphs** – provides information on how to build simple graphs using the SDK filters.
- **Section 5: Sample Applications** – describes samples included in the Elecard DirectShow Codec SDK.

1.1.3 Related Documentation

In order to thoroughly understand the DirectShow® technology, we strongly recommend that you read the following documentation:

- **Microsoft® DirectShow® Programmer Reference** described in Microsoft® DirectX Software Development Kit. You can find these documents at: <http://msdn.microsoft.com>.

1.2 Preface

The Elecard DirectShow Codec SDK is a software development kit intended to enable programmers to develop digital video transcoding applications using the Elecard components within the Microsoft® DirectShow® technology.

The Elecard DirectShow Codec SDK package includes the following:

1.2.1 Documentation

Elecard DirectShow Codec SDK documentation – consists of the following documents:

- Elecard DirectShow Codec SDK Reference Manual (this document)
- Elecard DirectShow Codec SDK Release Notes
- Elecard Components Reference Manuals
- Elecard Module Configuration Programmer Guide

1.2.2 Components

This section provides a quick overview of the DirectShow filters and other components included in SDK package. For further details, see the Elecard Components Reference documentation.

Table 1. Elecard DirectShow Codec SDK Components

Component	Description	File Name
Elecard MPEG-2 Video Decoder	DirectShow filter for software-only decoding MPEG-2 video (ISO/IEC 13818-2) and MPEG-1 video (ISO/IEC 11172-2) streams.	em2vd.ax
Elecard MPEG-2 Video Encoder	DirectShow filter for video encoding into MPEG-2 (ISO/IEC 13818-2) streams.	em2venc.ax
Elecard VPX Video Decoder	DirectShow filter for software-only decoding of VP8 and VP9 video streams. The component is based on/contains/uses libvpx.	evpxdec.ax
Elecard MPEG Audio Decoder	DirectShow filter for the software-only decoding of MPEG-1, MPEG-2, MPEG-2.5 and LPCM audio streams.	emad.ax
Elecard MPEG Audio Encoder	DirectShow filter for audio encoding into MPEG streams.	emac.ax
Elecard AAC Audio Decoder	DirectShow filter for the software-only decoding of AAC and HE-AAC audio streams.	eaacd.ax
Elecard AAC Audio Encoder	DirectShow filter for audio encoding into AAC streams.	eaace.ax
Elecard AVC Video Decoder	Software-only decoding solution for ISO/IEC 14496 part 10 AVC / ITU-T Recommendation H.264 video streams.	eavcdec.ax
Elecard AVC Video Encoder	Software module for video encoding into AVC/H.264 (MPEG-4 Part 10, ISO/IEC 14496-10) streams.	eavcenc.ax
Elecard HEVC Video Decoder	Software-only decoding solution for ISO/IEC 23008-2 MPEG-H Part 2/ITU-T H.265 video streams.	ehevdec.ax
Elecard HEVC Video Encoder	DirectShow filter for video encoding into ISO/IEC 23008-2 MPEG-H Part 2/ITU-T H.265 streams.	ehevenc.ax
Elecard QS AVC Video Decoder*	DirectShow filter designed for decoding ISO/IEC 14496 part 10 AVC / ITU-T Recommendation H.264 video and based on Intel QuickSync technology.	eqsdec_avc.ax
Elecard QS AVC Video Encoder*	DirectShow filter using Intel QuickSync for video encoding into AVC/H.264 (MPEG-4 Part 10, ISO/IEC 14496-10).	eqsenc_avc.ax
Elecard QS MPEG-2 Video Decoder*	DirectShow filter designed for decoding MPEG-2 video (ISO/IEC 13818-2) streams and based on Intel QuickSync technology.	eqsdec_mp2.ax
Elecard QS MPEG-2 Video Encoder*	DirectShow filter using Intel QuickSync technology for encoding into MPEG-2 video (ISO/IEC 13818-2).	eqsenc_mp2.ax
Elecard QS HEVC Video Decoder*	DirectShow filter designed for decoding ISO/IEC 23008-2 MPEG-H Part 2 / ITU-T H.265 video streams and based on Intel QuickSync technology.	eqsdec_hevc.ax
Elecard QS HEVC Video Encoder*	DirectShow filter designed for video encoding into H.265/HEVC (ISO/IEC 23008-2 MPEG-H Part 2/ITU-T and based on Intel QuickSync technology.	eqsenc_hevc.ax
Elecard QS VPP*	Video post processing DirectShow filter based on Intel QuickSync technology.	eqsvpp.ax
Elecard HLS Source	DirectShow source filter that provides connection to a server via HTTP Live Streaming (HLS) protocol version 3 and receives video/audio data from the server.	ehlss.ax
Elecard HLS Sink	DirectShow filter that is intended for content preparation via Apple HTTP Live Streaming (HLS) protocol.	ehlssink.ax

Component	Description	File Name
Elecard NWSource-Plus	DirectShow filter for receiving media data from the network. It receives the RTP and UDP packets and feeds the filter graph with stream data contained in these packets.	enwsplus.ax
Elecard RTSP NetSource	DirectShow filter that sets the connection with RTSP server sends request for starting, stopping, pausing the media broadcasting and positioning in the stream, receives media data from RTSP server.	ertspnws.ax
Elecard Sink Filter	DirectShow filter that dumps a stream to a file. It supports the indexing of the MPEG-2 VES and MPEG-2 PS and splitting the encoded file into parts of a predefined size.	esf.ax
Elecard NWRenderer	DirectShow filter for broadcasting media data to the network. It is capable of sending RTP, UDP and TCP packets and supports the announcement of its data session via SAP (SDP) packets.	enwr.ax
Elecard TimeMarker	DirectShow transform filter for stream analysis. It defines the type of data, extracts time stamps from the stream and sets these time stamps on the output media samples.	ETimeMarker.ax
Elecard StreamPump	DirectShow filter for converting asynchronous source to synchronous.	estrp.ax
Elecard MPEG Demultiplexer	DirectShow filter for splitting of MPEG-1 System Streams (ISO/IEC 11172-1), MPEG-2 Program and Transport Streams (ISO/IEC 13818-1) into video and audio streams.	empgdmx.ax
Elecard MPEG Multiplexer	DirectShow filter that provides the MPEG-2 Transport Stream (TS) or MPEG-2 Program Stream (PS) generation.	empegmux.ax
Elecard MPEG Push Demultiplexer	DirectShow filter for the software-only splitting of MPEG-1 System Streams, MPEG-2 Program Streams and MPEG-2 Transport Streams into video and audio streams.	empgpdmx.ax
Elecard MP4 Demultiplexer	DirectShow filter that provides demultiplexing of ISO/IEC 14496-14 file format (MP4) and 3GPP2 System streams into a MPEG-4, H.263, AVC/H.264 video streams and AAC, AMR, MPEG-1/2 Audio Layer 3 audio streams.	emp4dmx.ax
Elecard MP4 Multiplexer	DirectShow filter intended for the generation of MPEG-4 (Intermedia Format (MP4)) System Streams.	emp4mux.ax
Elecard LATM Demultiplexer**	DirectShow filter that provides demultiplexing of AAC LOAS/LATM stream into elementary AAC streams.	elatmdmx.ax
Elecard MKV Demultiplexer	DirectShow filter that provides demultiplexing MKV/WebM files. This component based on/contains/uses libmatroska and libebml.	emkvdmx.ax ebml_dmx.dll matroska_dmx.dll
Elecard MKV Multiplexer	DirectShow filter that provides the MKV multiplexing. This component based on/contains/uses libmatroska and libebml.	emkvmux.ax ebml_mux.dll matroska_mux.dll
Elecard MXF Demultiplexer	DirectShow filter designed for the software-only splitting of Material Exchange Format (MXF) into video, audio and ancillary data streams. The component is based on/contains/uses libmxf.	emxfdmx.ax
Elecard MXF Multiplexer	DirectShow filter intended for the generation of Material eXchange Format (MXF) streams. The component is based on/contains/uses libmxf, libmxf++, bmx, uriparser.	emxfmux.ax
Elecard Graph Viewer	Elecard Graph Viewer is a utility for the presentation of graphs built by any application. Elecard Graph Viewer allows the viewing and changing the filter properties, building of the filter graph (filter adding, deleting and connection), controlling of the graph state (run, stop, pause) and positioning in the media stream.	EIGViewer.dll
Elecard InfTee	DirectShow filter that delivers data (samples) received by its input	einftee.ax

Component	Description	File Name
	pin to an infinite number of output pins.	
Elecard File List Source**	DirectShow source filter that provides operations with a list of source files as with a single file.	efls.ax
Elecard Color Space Converter 2	DirectShow filter that provides conversion between different formats of uncompressed video.	ecsc2.ax
Elecard Module Config Checker	DirectShow filter that provides activation of the Elecard components in the GraphEdit application (contained in the DirectShow package).	echecker.ax
Elecard Desktop Capture	DirectShow filter that provides capturing of the specified screen area.	edc.ax

* 64-bit components only.

** 32-bit components only.

1.2.3 DirectShow Base Classes

DirectShow Base Classes are a set of C++ classes and utility functions designed for implementing DirectShow filters. Several of the helper classes are also useful for application developers. DirectShow Base Classes are included in Elecard DirectShow Codec SDK as its components and provided under the license "License\Microsoft IP Rights Notices.rtf"

1.2.4 Elecard Base Classes

Elecard Base Classes – a C++ class library that simplifies common tasks, appearing during development of multimedia applications, such as: DirectShow graphs building, filters and pins control etc. Base classes are used in SDK sample applications and are delivered in source form.

1.2.5 Sample Applications

The Elecard DirectShow Codec SDK samples are simple applications that demonstrate the Elecard components and Base Classes use. The samples are written in C++. The following table provides a brief overview of each sample. For further details, see Section Sample Applications.

Table 2. Elecard DirectShow Codec SDK Sample Applications

SDK samples	Sample Name	Description	File Name
Decoders	Simple Player	Sample application that demonstrates use and adjustment of the Elecard Video Decoders.	SimplePlayer.exe
	Console Player	Sample console application that demonstrates use and adjustment of the Elecard decoders without use of Elecard Base Classes.	ConsolePlayer.exe
Encoders	Simple Encoder	Sample application that demonstrates use and adjustment of the Elecard Encoders and multiplexers.	SimpleEncoder.exe
	Console Encoder	Sample console application that demonstrates use and adjustment of the Elecard encoders without the Elecard Base Classes use.	ConsoleEncoder.exe
	Simple Capture	Sample application that encodes stream from a capture device (video – into AVC format, audio – into MPEG-1 or MPEG-2 Audio format), multiplexes it into MPEG-2 TS or PS format and dumps it to a file.	SimpleCapture.exe

SDK samples	Sample Name	Description	File Name
LowDelay	Low Delay Client	Sample application for receiving and playback of video and audio data that is captured and broadcast with Low Delay Streamer.	LowDelayClient.exe
	Low Delay Streamer	Sample application that captures video from the specified screen area and audio from the specified PC audio device, transcodes the data into selected formats, multiplexes the encoded streams into MPEG-2 Transport Stream and broadcasts the stream to network	LowDelayStreamer.exe
Network	NWPlayer	Sample application that plays media streams from the network and allows the received media data saving onto a disk. It supports the media files indexation using Sink Filter. This sample illustrates the usage of the Elecard Module Configuration technology and the usage and adjustment of the NWSource-Plus, RTSP NetSource and HLS Source filters.	NWPlayer.exe
	NWServer	Sample application that demonstrates broadcasting media data to the network. It is capable of sending RTP, UDP and TCP packets and sends data session announcement via SAP (SDP) packets.	NWServer.exe
	NWServerTranscoder	Sample application that demonstrates transcoding and broadcasting media data to the network. The broadcasting capabilities are the same as in the NWServer application.	NWServerTranscoder.exe
QuickSync	Simple Player QS	Sample application that demonstrates use the Elecard QS Video Decoders and Elecard QS VPP for files playback.	SimplePlayerQS.exe
	Simple Encoder QS	Sample application that demonstrates use the Elecard QS Video Decoders and Encoders for transcoding tasks.	SimpleEncoderQS.exe

1.3 Requirements

The Elecard DirectShow Codec SDK has the following hardware and software requirements:

1.3.1 Hardware Requirements

Minimum hardware requirements if using software Elecard components:

- 2 GHz 32-bit (x86) or 64-bit (x64) processor
- 1 GB RAM (32-bit) or 2 GB RAM (64-bit)
- Any VGA card

Note: For Windows versions earlier than Vista, audio capturing is performed, if the system sound card supports Stereo mixer.

Minimum hardware requirements if using Elecard QuickSync components:

- Intel® Xeon® Processor E3-1200 v4 series.
- Intel® Xeon® Processor E3-1200 v5.
- Intel® Xeon® Processor E3-1500 v5 series.
- 6th Generation Intel Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris™ Graphics or Intel HD Graphics (580, 540, 530).
- 5th Generation Intel Core™ Processors with Intel Iris™ Pro Graphics, Intel Iris™ Graphics or

Intel HD Graphics (5500, 6000, 6100, 6200).

Note: HEVC video decoding and encoding are supported only on Intel® 6th generation “Skylake” CPU.

1.3.2 Software Requirements

- Windows® 8/10
- Intel® Graphics driver, version 24.20.100.6025 (QuickSync components)
- CMake 3.6 or later

1.4 Technical Support

For technical support contact the Elecard Technical Support Team: tsup@elecard.com.

2. Getting Started

2.1 Introduction

The following section details the procedures for installing the Elecard DirectShow Codec SDK. In addition, it provides the description of the Elecard DirectShow Codec SDK folder structure.

Note: *The described installation/uninstallation process is relevant for all versions of the Elecard DirectShow Codec SDK.*

2.2 Installing Elecard DirectShow Codec SDK

To install the Elecard DirectShow Codec SDK:

1. Run the Elecard DirectShow Codec SDK setup. To run, double click the executable file from the Elecard DirectShow Codec SDK setup package.
2. The *Elecard DirectShow Codec SDK setup* window will appear. Read the recommendations and warnings. Click **Next**.
3. The Release Notes will appear. Click **Next**.
4. The license agreement will appear. Read the agreement and if you accept the terms within, select the “*Yes I agree with the terms of this license agreement*” check box. Click **Next**.
5. Select the destination folder in which you want to install the Elecard DirectShow Codec SDK. Click **Next**.
6. Select the program group in which you want the Elecard DirectShow Codec SDK to be located. Click **Next**.
7. To complete installation, follow the onscreen instructions. When setup has finished installing all of the necessary files on your computer, the appropriate message box with the text “*Elecard DirectShow Codec SDK has been successfully installed*” will appear and the SDK is ready to use.

2.3 Uninstallation Elecard DirectShow Codec SDK

To uninstall the Elecard DirectShow Codec SDK application:

1. Click *Start*→*Programs*→*Elecard*→*Elecard DirectShow Codec SDK*→*Uninstall*.
2. Follow the onscreen instructions to complete removal of the application.

2.4 Describing Codec SDK Folder Structure

After installing the Elecard DirectShow Codec SDK, the *Elecard DirectShow Codec SDK* folder will appear in the destination folder specified during installation.

The SDK folder contains:

- **Bin** – contains the executable binaries for the SDK samples
- **Components** – contains the Elecard DirectShow filters and engine DLLs
- **Doc** – includes all SDK-related documentation
- **License** – includes the third-party licenses
- **Sources** – contains the source codes:
 - **BaseClasses** – includes the source codes of the Elecard DirectShow Codec SDK base class library
 - **DirectShow\BaseClasses** – includes the source codes of the Microsoft® DirectShow® base

- class library
- **Components** – contains header files (*.h files, *.inc files)
- **Samples** – contains the source codes of the Elecard DirectShow Codec SDK sample applications

2.5 DirectShow Filters Activation

Most of the encoder and decoder (video and audio) filters from the SDK have a copy-protection mechanism: without activation these filters operate in an evaluation mode (e.g. overlay logo on the video). After activation filters operate in a demo mode (e.g. still overlay logo on the video, but without restrictions imposed on an expired mode). When the development is finished, the OEM pack with the components used in the product must be ordered (licensed) from Elecard.

There are several ways to perform activation:

- Activation via *Registrator* using a special activation number. In this case every end-user should start the *Registrator* application and type the unique activation number. This activation method is intended for end-users.
- "Instance Activation" from an application without any additional tools (without *Registrator*). This activation method is intended for OEM customers that distribute filters within their own applications.
- Activation with the Elecard Module Config Checker filter using a special GUID. This activation method is intended for tests using the GraphEdit or GraphStudioNext applications. To utilize the Elecard Module Config Checker features, insert the filter into your filter graph, open its property page, select components for activation, and type an activation GUID.

Choosing the activation way depends on the development project requirements.

If it is required to redistribute the application and the codec plug-in separately, then it is reasonable to get the OEM pack as a special Installer with built-in Registrator. Certain parameter presets can be included as the default for each filter. So, the application does not need to reconfigure components. The package needs to be installed and activated by the end-user or by the application ('silent' installation and activation are available when you pass `-var:"SilentMode=1"` and `-var:"SerialNumber=xxxxxxxxxxxxxxxx"` as the Installer command-line parameters). Serial number is provided.

If the Elecard components must be included into the application install-pack, then it is reasonable to get the OEM pack as the filter binaries (*.ax files). When the application is installed, the binaries need to be placed to the known location on target computer. The filters can be registered in DirectShow environment and loaded in standard way. It is possible to use components without registration, but the only way to activate them is to use "Instance Activation" by a `KEY_GUID` inside the application. The `KEY_GUID` which is provided with the OEM pack is valid for all the filters in the package.

The direct filter usage without registration (is implemented in all SDK samples) is the most preferable from the security and stability point of view. Even if some other Elecard components are installed/uninstalled on target computer, the application uses its own set of binaries and does not depend on the current DirectShow environment.

2.5.1 Instance Activation

In the case of "Instance Activation" an application should pass a special GUID named `KEY_GUID` into the filter.

`KEY_GUID` is unique for every OEM customer and common for all filters supplied to him. Every filters update will be made with this `KEY_GUID`.

There is the sample code that should be integrated in the application for the filter activation.

```
HRESULT CBaseEncoder::ActivateFilter(CFilterWrapper* filter)
{
    HRESULT hr = E_FAIL;
    IModuleConfig* IMC;
    hr = filter->QueryInterface(&IMC);
    if (SUCCEEDED(hr))
    {
        IMC->SetValue(KEY_GUID, NULL);
        IMC->Release();
    }
    return hr;
}
```

Where KEY_GUID is defined as:

```
static const GUID KEY_GUID = { 0x00000000, 0x0000, 0x0000, { 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00 } };
```

Zeros should be replaced by KEY_GUID assigned to a certain OEM customer.

2.5.2 Adding a DirectShow Filter to the Graph without Registration

Please note that placing files into the shared components folder is potentially unsafe. If the user installs an application which contains the same modules as your application, the original files can be overwritten. So, new components with other KEY_GUIDs will not be activated by your application any more.

The most reliable method of preventing the modules collision is using the particular folder for the filters supplied with your application.

It is recommended to insert filter in the graph without its registration in the Windows (without using *regsvr32*).

The following code example shows how you can create a filter using the file path.

```
CFilterWrapper *m_encoder = NULL;

m_encoder = CFilterWrapper(CLSID_Encoder, A2OLE(m_EncoderPath));
if (m_encoder)
    m_hrError = AddFilter(m_encoder, _T("Video Encoder"));
```

where m_EncoderPath is path to encoder file.

This method of filters creation ensures the use of the appropriate filters.

3. DirectShow Overview

3.1 Introduction

The following section provides an overview of the DirectShow components, including filter types and data flow graphs.

3.2 DirectShow main terms and definitions

At the heart of **DirectShow®** is a modular system of pluggable components called *Filters*. These filters are arranged in a configuration called a *Filter Graph*. The *Filter Graph Manager* component oversees the connection of these filters and controls the data flow stream. Applications that use DirectShow architecture control the activities of the Filter Graph by communicating with the Filter Graph Manager.

Most of the filters included in Microsoft® DirectShow® runtime reside in quartz.dll, while others are standalone .AX files. The DirectShow filters are DLL files. Many of them are installed to Windows\system directory and others are installed in the products specific catalogs (for example, the Elecard filters are installed in the *Program Files\Elecard\<Product name>\Components* folder on the system disk).

The following is important to know about filter data streaming, connections, and pins:

- Data streams in packets (called *MediaSample*) from Source to Renderer through Transform filters.
- The connection from one filter to another is realized with the help of a *Pin*. A pin is an object that belongs to the filter. It provides a connection to other filter pins. The *Input Pin* receives a *MediaSamples* from upstream filter and the *Output Pin* sends the *MediaSamples* to downstream filter.
- The Source filter has at least one output pin, the Renderer filter has at least one input pin, and the Transform filter has both input and output pins.

3.2.1 DirectShow Filter Types

DirectShow filters have the following three main classes:

- **Source Filter** – provides the multimedia stream. It ranges from the File Source Filter to the MPEG encoding device source, or Network source.
- **Renderer Filter** – finishes a graph. The most common Renderer filters are Video Renderer and Audio Renderer, which play video and audio streams. A Renderer filter can also be a File Dump or File Writer filter and a Network Render filter.
- **Transform Filter** – the widest range of filters. All transformations on DirectShow® streams are made in transform filters. This type of filter is divided into Transform and TransInPlace filters. The TransInPlace filter differs from the Transform filter in memory allocation. It does not provide its own allocators, but uses ones from the upstream or downstream filter. It sends the same *MediaSample* that it received from the upstream to downstream filter and makes all data transformations in the *MediaSample* data buffer without changing its size.

3.2.2 Data Flow Principle

The following describes the DirectShow data flow principle using the example of an audio stream being split from an MPEG stream and then dumped to a file.

For this task realization Filter Graph will be consist of 3 following filters:

- **Async File Source** – gets data from the MPEG-1 file.

- **MPEG-1 Splitter** – splits MPEG-1 streams into MPEG-1 video and MPEG-1 audio elementary streams.
- **File Dump Filter** – writes compressed audio to the hard drive.



Figure 1. Audio Stream Splitting and Dumping to a File

Once the Audio file is split from the MPEG Stream and dumped into a file, it can easily be raised from the file, parsed, decoded and then played back by the following graph:

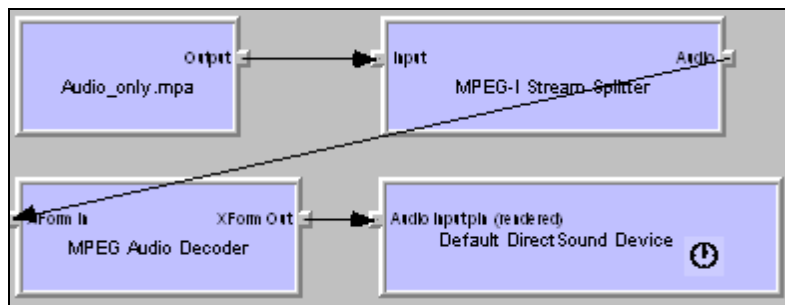


Figure 2. Dumped Audio File Playback Filter Graph

3.3 DirectShow Utility Applications

For visual representation of DirectShow graphs, use the GraphEdit utility, provided with Microsoft WindowsSDK 7.1, or use the open source [GraphStudioNext](#) utility.

Note: Microsoft WindowsSDK 7.1 is the last version supporting DirectShow, later versions of SDK do not contain base classes, application examples and utilities applied for DirectShow.

GraphEdit and GraphStudioNext allow:

- defining what components are integrated in the system,
- receiving CLSID of each component and location of a binary file,
- visualizing a graph building process (addition and connection of the components), receiving information on connection media type (data presentation format),
- receiving an access to the components settings using Property Pages,
- controlling data processing (to start/pause/stop a graph/change a position and etc.),
- and others.

To build simple graphs in GraphEdit using Elecard DirectShow Codec SDK filters, see instructions below.

3.3.1 Building a simple file playback graph using Elecard components

This example demonstrates building of the DirectShow graph for playback of an MPEG-1/MPEG-2 media files using the standard File Source filter and other Elecard filters such as: Elecard MPEG Demultiplexer, Elecard MPEG-2 Video Decoder, Elecard MPEG Audio Decoder.

Note: GraphEdit and GraphStudioNext can build a custom filter graph, using any of the filters registered on your system. The components provided in a set with Elecard DirectShow Codec SDK, are not registered in a user's system during installation. To register 32-bit Elecard components, run RedFilters.bat from the folder <Elecard DirectShow Codec SDK folder>\Components. 64-bit components and their RegFilters.bat are located in <Elecard DirectShow Codec SDK

folder>\Components\x64. Be advised, that BAT file should be run with administrative permissions.

To build the simple graph:

1. Start GraphEdit from the installed Microsoft WindowsSDK 7.1 (*..\Program Files\Microsoft SDKs\Windows\<version>\Bin\graphedt.exe or ..\Program Files\Microsoft SDKs\Windows\<version>\Bin\x64\graphedt.exe*).
2. On the **Graph** menu, click **Insert Filters**.
3. From the “Which filters do you want to insert?” window, open DirectShow Filters, choose **File Source (Async.)** and click **Insert Filter**.

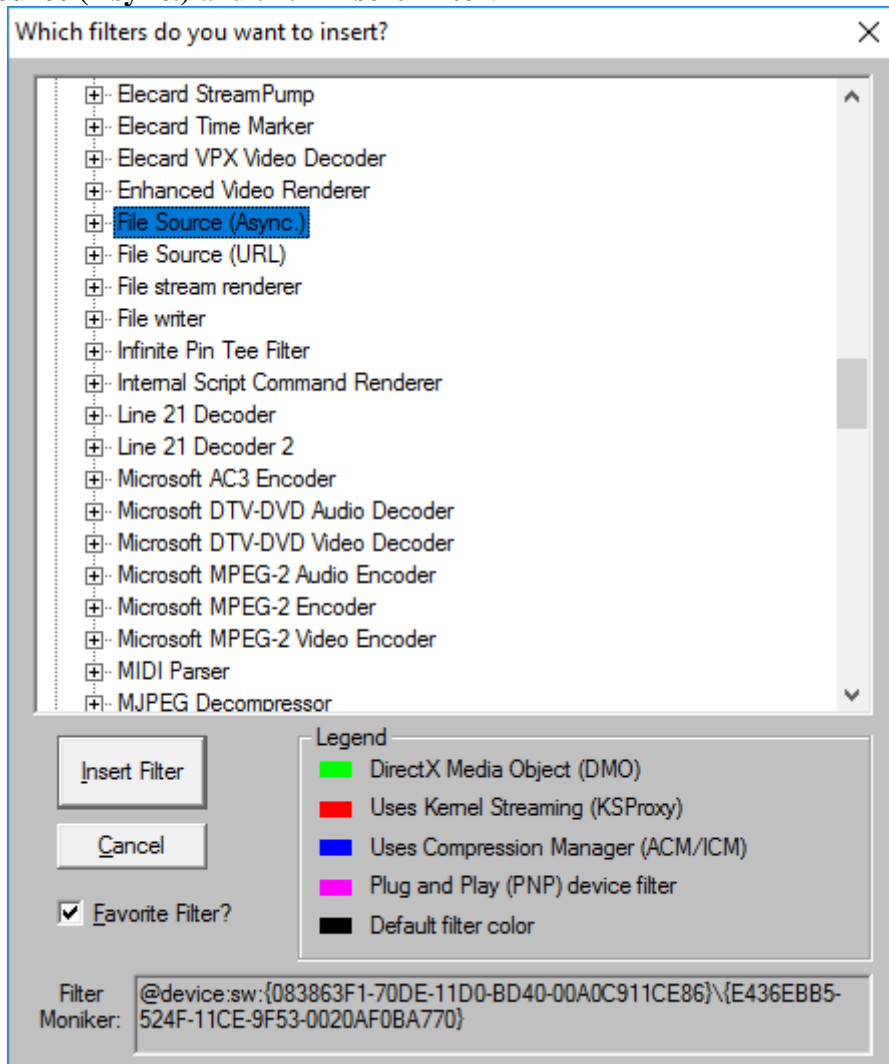


Figure 3. GraphEdit ‘Insert Filters’ Dialog

4. The “Select an input file...” window will appear. Choose the MPEG TS file you want to render.

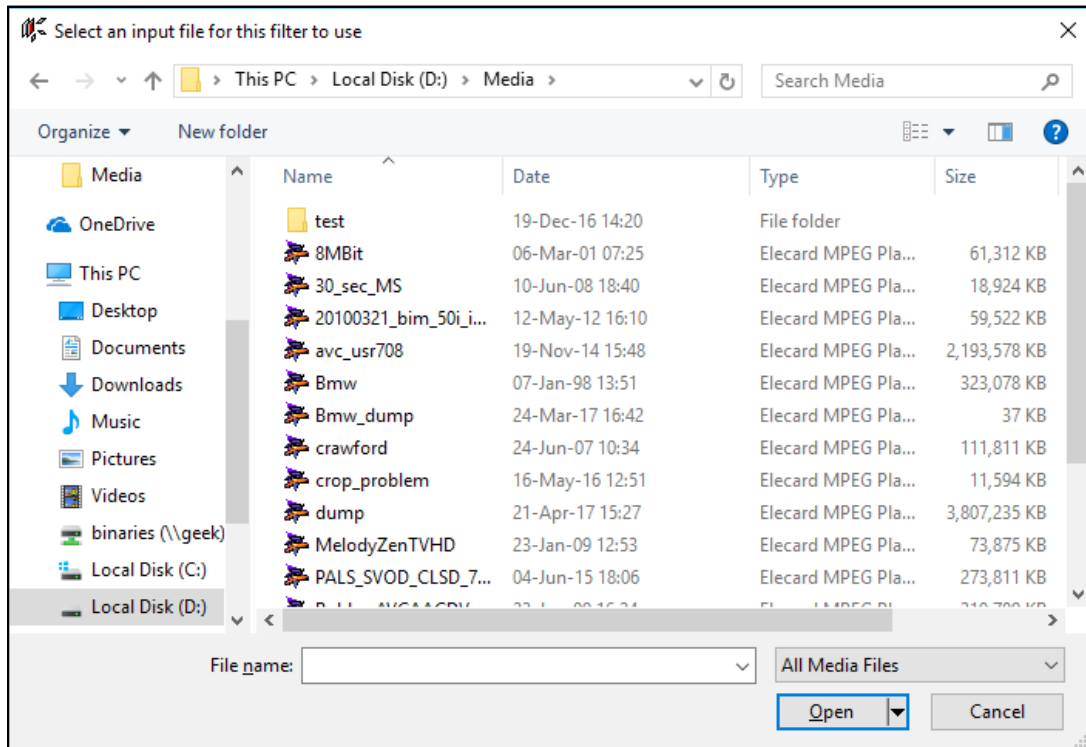


Figure 4. File Source (Async.) 'Select an input file...' Dialog

5. Insert the following DirectShow filters into the graph (as described above in Step 3): Elecard MPEG Demultiplexer, Elecard MPEG-2 Video Decoder, Elecard MPEG Audio Decoder.
6. Connect the File Source output pin with the Elecard MPEG Demultiplexer input pin.
7. Then, connect the Elecard MPEG Demultiplexer video pin with Elecard MPEG-2 Video Decoder input pin.
8. To create an automatic connection between Audio and Video Renderers, right-click the output pins of the Audio and Video Decoders and choose **Render** menu items. The following graph will appear:

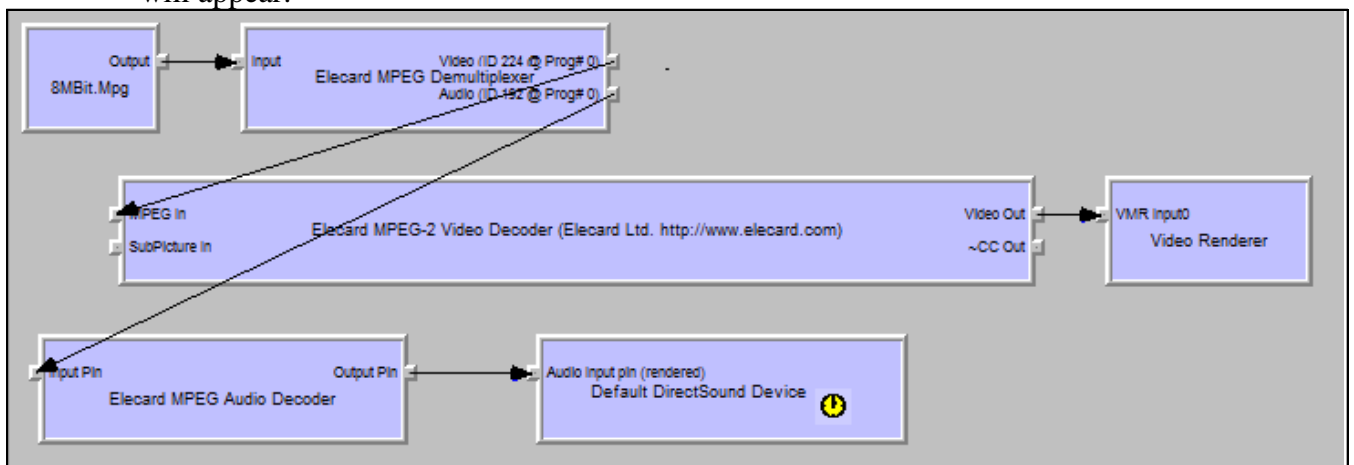


Figure 5. Simple Player Filter Graph

9. To start playback, on the Graph menu, click Play.

Note: For additional information on working with GraphEdit see the GraphEdit help. <https://msdn.microsoft.com/en-us/library/windows/desktop/dd407274.aspx>

4. Sample Applications

4.1 Introduction

This section describes the sample applications included in the Elecard DirectShow Codec SDK package. They are available from the SDK program group (*Start*→*Programs*→*Elecard*→*Elecard DirectShow Codec SDK*→*Samples*).

4.1.1 Building sample applications

Sample applications are CMake-based C++ applications.

[Visual Studio 2017](#) provides built-in support for handling CMake projects, simply open the folder (*Elecard DirectShow Codec SDK root*) in Visual Studio (via **File** > **Open** > **Folder...** or **devenv.exe** <*Elecard DirectShow Codec SDK root*>).

If you use earlier versions of Visual Studio (VS), please install [CMake](#) generate VS projects and solutions by means of **cmake-gui.exe** or call **cmake.exe** from a command prompt.

To create a solution using **cmake-gui.exe**, perform the following actions:

1. Launch **cmake-gui.exe**.
2. In the **Where is the source code** field set the following path → (*Elecard DirectShow Codec SDK root*)\Sources.
3. In the **Where to build the binaries** field set the path for configuration files to be created, for example: (*Elecard DirectShow Codec SDK root*)\Sources\build.
4. Click **Configure** button.
5. Select the installed version of Visual Studio for creating a solution file. Choose **Win64** version if need and click **Finish**.
6. Wait for the configuration stage completion and then click **Generate**.
7. Go to the directory you created in step 3.
8. Open **elecard_directshow_codec_sdk.sln** in Visual Studio IDE.
9. In the **Solution Explorer** tab right-click on **ALL_BUILD** and choose **Build** or press **F7**.
10. To create a solution using **cmake.exe**, it is necessary to perform the following actions in the command prompt:

- Go into the (*Elecard DirectShow Codec SDK root*)\Sources directory

```
cd (Elecard DirectShow Codec SDK root)\Sources
```

- Create a **Build** directory and move into it:

```
mkdir build
```

```
cd build
```

- To create a solution, run the following command:

```
cmake -G "Visual Studio 12 2013 Win64".
```

This is an example of a command to create a solution file using **cmake 3.7** and **Visual Studio 2013 x64**. If you need to create a solution for the other Visual Studio version, change the command.

- Complete the steps **8** and **9** described above or run the following command:

```
cmake --build . --config Debug --clean-first
```

4.2 Elecard DirectShow Codec SDK Sample Applications

The following describes the SDK Samples, including:

- Simple Player
- Simple Player QS
- Console Player
- Simple Encoder
- Simple Encoder QS
- Console Encoder
- Simple Capture
- Low Delay Streamer
- Low Delay Client
- NWPlayer
- NWServer
- NWServerTranscoder

4.2.1 Simple Player

4.2.1.1 Description

Simple Player is a sample playback application that demonstrates use and adjustment of the Elecard decoders. The application represents basic player functionality (start, stop, fast forward, rewind, positioning, etc). **ModuleConfig** interface is used for the Elecard decoders parameters configuration.

The **Use indexing file** option allows creating and using indexing files for precise positioning of MPEG TS/PS streams.

Depending on a file format, some of GUI elements may differ or be unavailable. For example, **Use DXVA 2.0** is not available while playing video in HEVC and VP8/VP9.

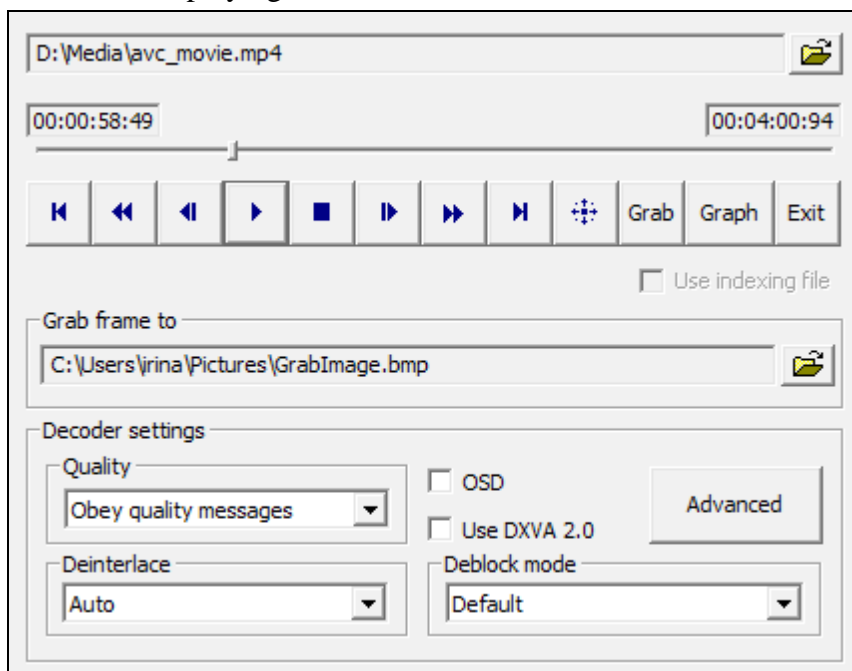


Figure 6. Simple Player GUI Example

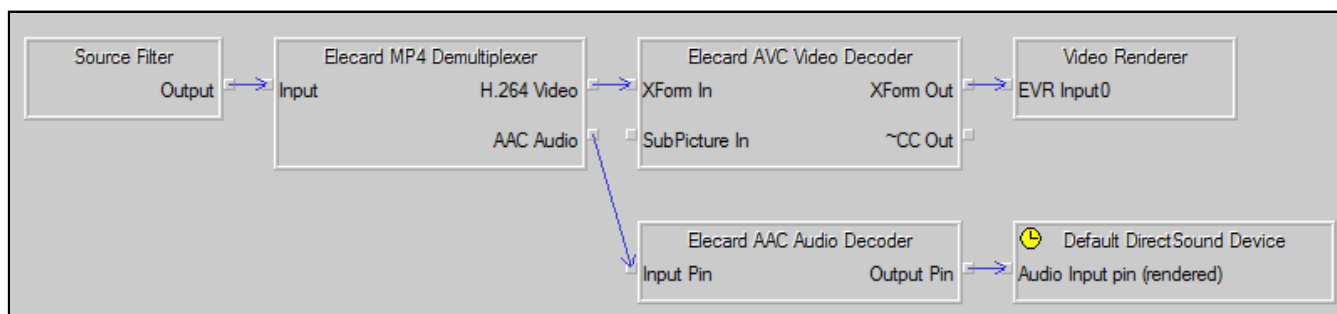


Figure 7. Filter Graph Example

4.2.1.2 Path

- Source** (Elecard DirectShow Codec SDK root)\Sources\Samples\Decoders\SimplePlayer
- Binaries** (Elecard DirectShow Codec SDK root)\Bin\SimplePlayer.exe
(Elecard DirectShow Codec SDK root)\Bin\x64\SimplePlayer.exe

4.2.1.3 Features

Simple Player implements the following features:

- Software decoding VP8/VP9, AVC, HEVC, MPEG-2 video using Elecard components
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Adjustment of base parameters of Elecard video decoders using the Player main window controls
- Access to all video decoders settings using the **Advanced** button
- Current frame saving to file
- Support of index file creation and loading (for MPEG TS only)
- The built graph visualization using the Elecard Graph Viewer COM object

4.2.2 Simple Player QS

4.2.2.1 Description

Simple Player QS is a sample application similar to Simple Player, but only Elecard QS decoders are applied for decoding. The sample can also demonstrate operation of the Elecard QS VPP component, that allows performing post processing transformations (e.g. scaling, deinterlacing, frame rate changing).

Be advised, that:

- only 64-bit versions of QuickSync components are included in Elecard SDK,
- to ensure proper operation of the sample, make certain that your computer complies with the requirements stated in the Section 1.3.

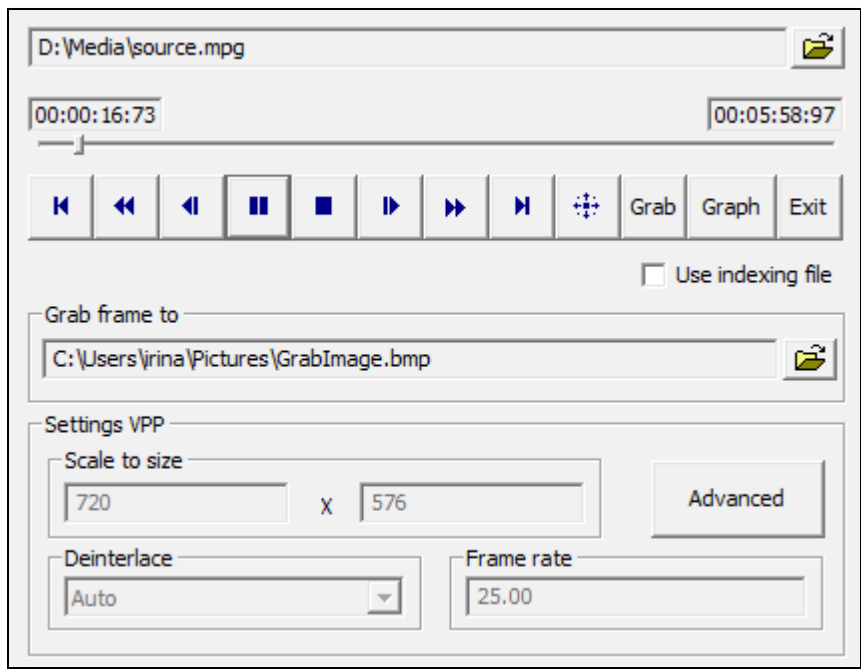


Figure 8. Simple Player QS GUI Example

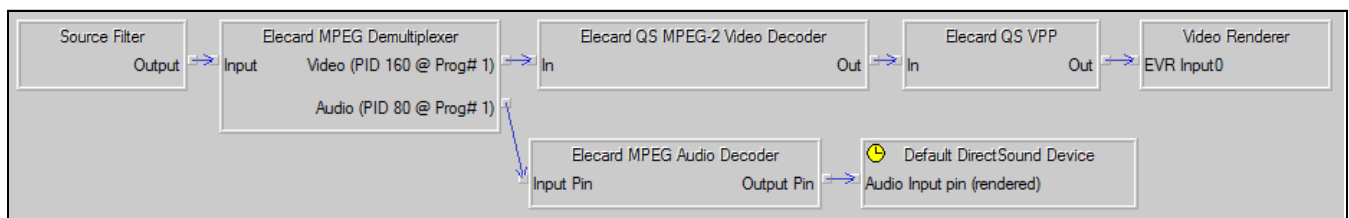


Figure 9. Filter Graph Example

4.2.2.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\QuickSync\SimplePlayer
Binaries (Elecard DirectShow Codec SDK root)\Bin\x64\SimplePlayerQS.exe

4.2.2.3 Features

Simple Player QS implements the following features:

- Hardware decoding AVC, HEVC, MPEG-2 video using Elecard QS Video Decoders
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Adjustment of base parameters of Elecard QS VPP using the Player main window controls
- Current frame saving to file
- Support of index file creation and loading (for MPEG TS only)
- The built graph visualization using the Elecard Graph Viewer COM object

4.2.3 Console Player

4.2.3.1 Description

Console Player is a sample console application that demonstrates use and adjustment of the Elecard software decoders without use of Elecard Base Classes. The application work is controlled using command line:

```
ConsolePlayer.exe <Input file> [options]
```

Note: Input file is the mandatory parameter and specifies the path of a media file for playback.

Options are a set of optional parameters. The following options can be set when working with Console Player:

- --dxva – enables the hardware acceleration;
- --osd – enables the displaying of internal information about decoding process;
- --d <value> - specifies decoder deinterlacing mode. Available values:
 - 0 – perform deinterlacing on interlaced frames and does not deinterlace progressive frames;
 - 1 – perform deinterlacing on all types of frames;
 - 2 – do not perform deinterlacing;
 - 3 – generate one progressive frame from every field;
 - 4 – generate interlaced frames with doubled frame rate;
- --q <value> - specifies types of frames that decoder will skip. Available values:
 - 0 – respect quality messages from upstream filter and choose frame types to skip automatically to maintain synchronized playback;
 - 1 – skip P and B frames even if they are used as reference;
 - 2 – skip B frames even if they are used as reference;
 - 3 – decode all frames, do not skip;
 - 4 – skip all non-reference frames.

Examples of usage:

```
Console_Player.exe D:\Media\MediaFile.mpg
```

```
Console_Player.exe D:\Media\MediaFile.mpg --dxva --q 3
```

```
Console_Player.exe D:\Media\MediaFile.mpg --dxva --q 3 --osd --d 2
```

Optional parameters can not be applied to all of the Elecard decoders.

Elecard MPEG-2 Video Decoder and Elecard AVC Video Decoder support all the options during the evaluation period. If your evaluation period is expired, you won't be able to use hardware acceleration when decoding (--dxva option won't work).

--dxva and --d options are not supported in Elecard HEVC Video Decoder.

Elecard VPX Video Decoder supports none of the defined options.

4.2.3.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\Decoders\ConsolePlayer

Binaries (Elecard DirectShow Codec SDK root)\Bin\ConsolePlayer.exe
(Elecard DirectShow Codec SDK root)\Bin\x64\ConsolePlayer.exe

4.2.4 Simple Encoder

4.2.4.1 Description

Simple Encoder is a sample application that demonstrates use and adjustment of the Elecard software components for media streams transcoding.

The application allows changing media data format of the source file without changing internal data formats (remultiplexing). To proceed, choose “Nothing encoder used” as a value for video and audio encoders.

If “Nothing encoder used” is chosen only for video or only for audio encoder, Simple Encoder will work as follows: first video or audio stream of the source file is decoded, encoded using selected Elecard encoder filter, multiplexes and saved to a destination file.

If video and audio encoders are set, application will function as a transcoder, i.e. it transcodes first video and audio streams of the source file by means of the chosen encoders, and multiplexes them by means of

the chosen muxer.

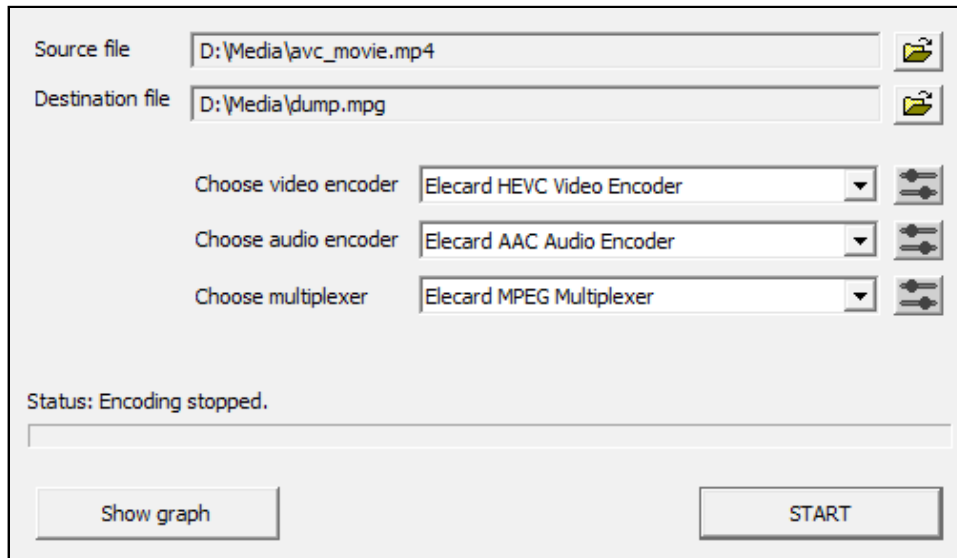


Figure 10. Simple Encoder GUI

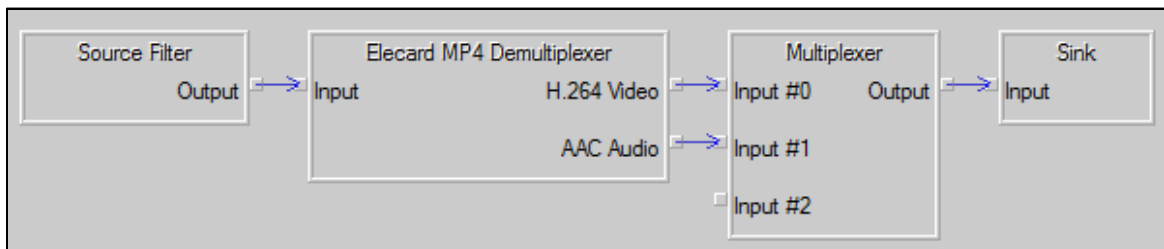


Figure 11. Simple Encoder Transcoding Filter Graph example

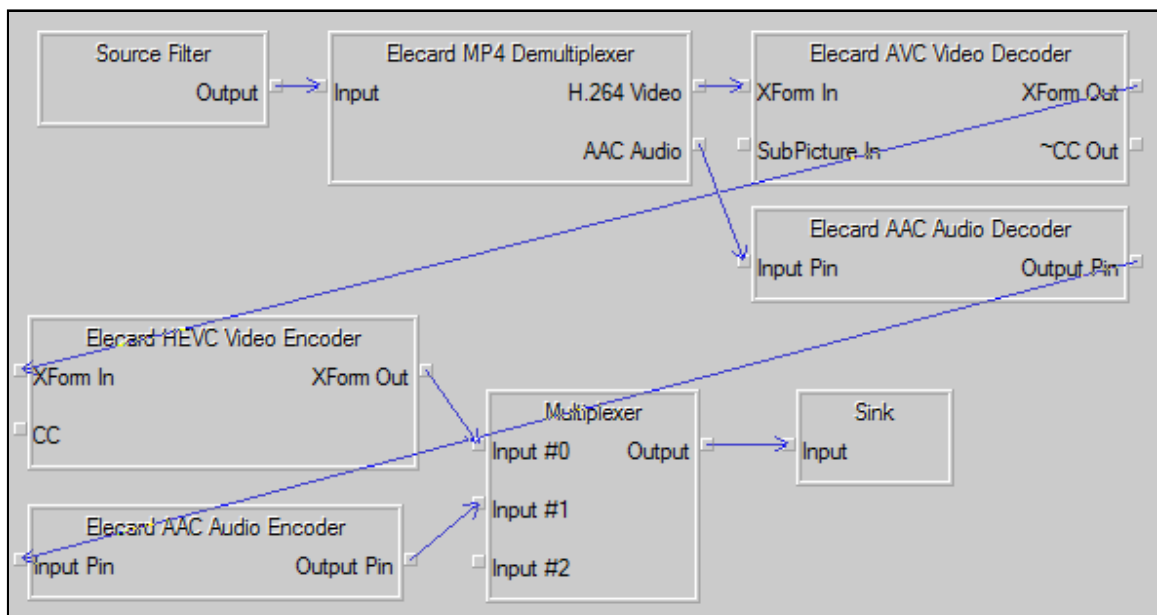


Figure 12. Simple Encoder Remultiplexing Filter Graph example

4.2.4.2 Path

- Source** (Elecard DirectShow Codec SDK root)\Sources\Samples\Encoders\SimpleEncoder
Binaries (Elecard DirectShow Codec SDK root)\Bin\SimpleEncoder.exe
 (Elecard DirectShow Codec SDK root)\Bin\x64\SimpleEncoder.exe

4.2.4.3 Features

Elecard Simple Encoder implements the following features:

- Encoding video into AVC, HEVC, MPEG-2 using software components
- Encoding audio into AAC, MPEG Audio
- Remultiplexing input stream with or without transcoding to MP4, MPEG-2, MKV, MXF
- Access to Elecard encoders and multiplexer settings
- The built graph visualization using the Elecard Graph Viewer COM object

4.2.5 Simple Encoder QS

4.2.5.1 Description

Simple Encoder QS – is a sample application that demonstrates use of the Elecard QS Video Decoders and Encoders for media streams transcoding. Simple Encoder QS is similar to Simple Encoder, but only Elecard QS components are applied for video coding.

Be advised, that:

- only 64-bit versions of QuickSync components are included in Elecard SDK,
- to ensure proper operation of the sample, make certain that your computer complies with the requirements stated in the Section 1.3.

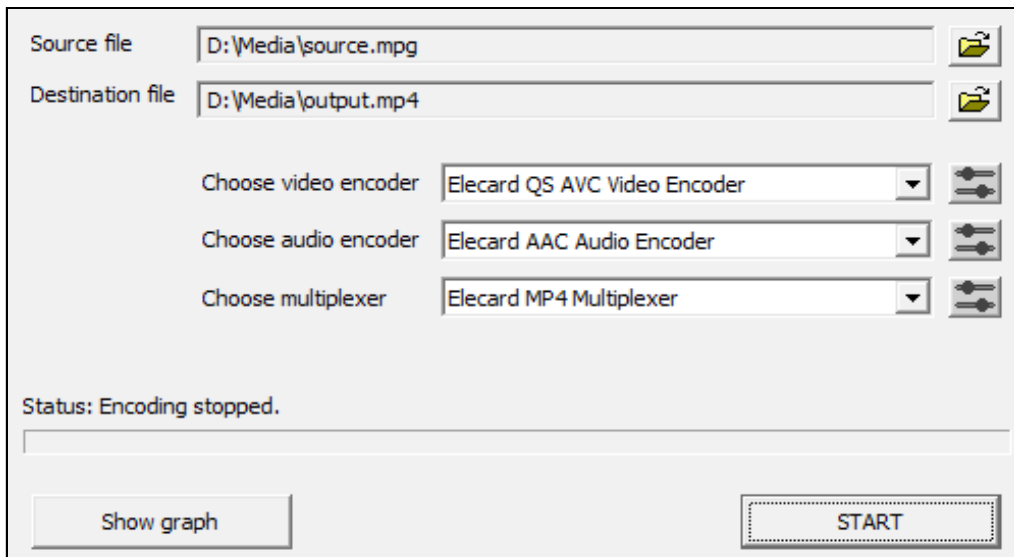


Figure 13. Simple Encoder QS GUI

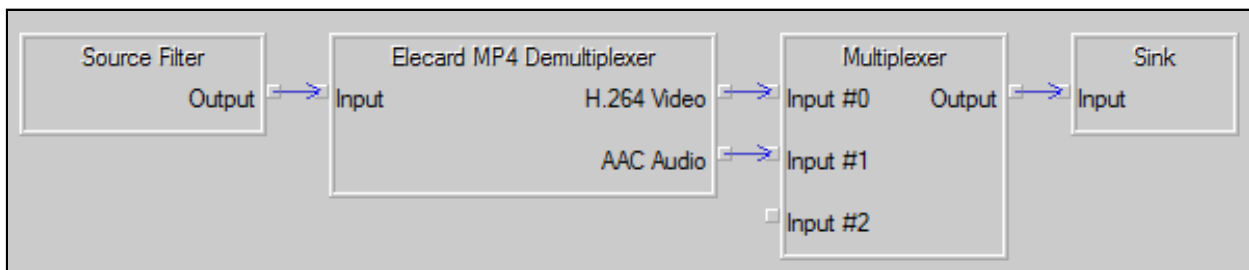


Figure 14. Simple Encoder QS GUI

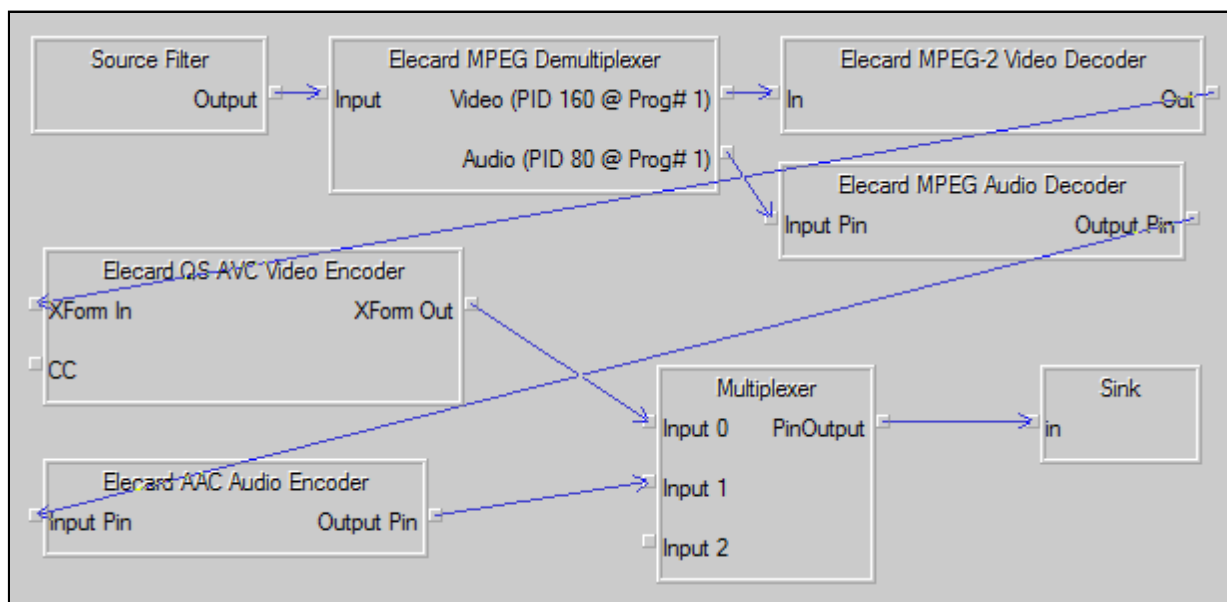


Figure 15. Simple Encoder QS Remultiplexing Filter Graph example

4.2.5.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\QuickSync\SimpleEncoderQS

Binaries (Elecard DirectShow Codec SDK root)\Bin\x64\SimpleEncoderQS.exe

4.2.5.3 Features

Elecard Simple Encoder QS implements the following features:

- Encoding video into AVC, HEVC, MPEG-2 using Elecard components supported Intel QuickSync
- Encoding audio into AAC, MPEG Audio
- Remultiplexing input stream with or without transcoding to MP4, MPEG-2
- Access to all Elecard encoders and multiplexer settings
- The built graph visualization using the Elecard Graph Viewer COM object

4.2.6 Console Encoder

4.2.6.1 Description

Console Encoder is a sample console application that demonstrates use and adjustment of the Elecard video software encoders without use of Elecard Base Classes.

The working scheme is the following: video stream of the source file is decoded and then encoded into specified format. If the source file contains audio data, the audio stream is ignored. The encoded stream dumped to file

The application work is controlled using command line:

```
ConsoleEncoder.exe <Input file> [options]
```

Input file – Specifies the path of media file for encoding. The mandatory parameter.

Options are a set of optional parameters. The following options can be set when working with Console Encoder:

- --enc <value> – specifies the video encoding format. Available values:
 - mpeg2,
 - avc,

- hevc (default)
- --mode – defines bitrate mode:
 - cbr – sets encoder to Constant Bitrate mode (for HEVC encoding Const Quantizer mode will be used)
 - vbr – sets encoder to Variable Bitrate mode.
- --avg <value> – specifies the video encoding average bitrate in bits per second.
- --max <value> – specifies the video encoding maximum bitrate in bits per second. This option has no effect if CBR mode is used.
- --out <output file> – specifies the path of file for encoded data saving. If it is not specified the encoded data is saved into *dump.mpg* file located in the same directory as ConsolePlayer.exe.

For example, to encode a video stream from the file *c:\video\input.mpg* to AVC/H.264, 15000000 bps, CBR use the following command line:

```
ConsoleEncoder.exe c:\video\input.mpg --enc avc --mode cbr --avg 2500000
```

The result should be recorded in the *dump.mpg* file.

4.2.6.2 Path

- Source** (Elecard DirectShow Codec SDK root)\Sources\Samples\Encoders\ConsoleEncoder
- Binaries** (Elecard DirectShow Codec SDK root)\Bin\ConsoleEncoder.exe
(Elecard DirectShow Codec SDK root)\Bin\x64\ConsoleEncoder.exe

4.2.7 Simple Capture

4.2.7.1 Description

Simple Capture is a sample application that encodes stream from a capture device (video – into AVC format, audio – into MPEG-1 or MPEG-2 Audio format), multiplexes it into MPEG-2 TS or PS format and dumps it to a file.

This sample demonstrates work with any capture devices installed in the system. Only one capture device is enabled at once. In the case of DV capturing the application provides the camcorder control (playback tart, stop and rewind).

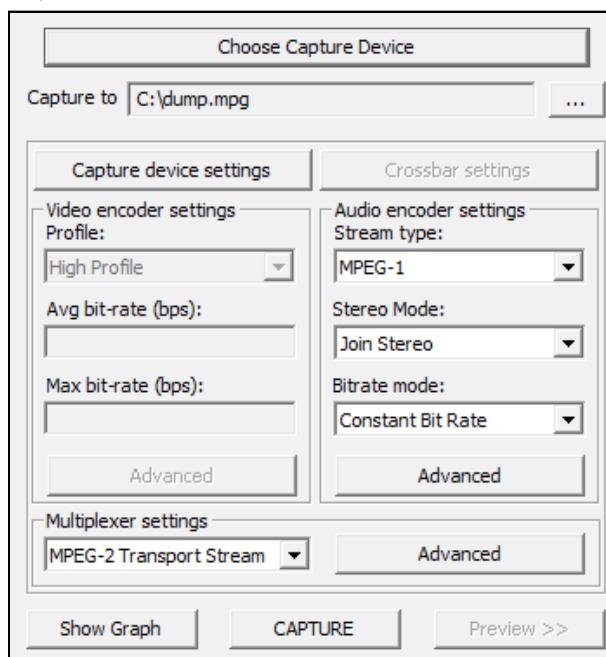


Figure 16. Simple Capture GUI

4.2.7.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\Encoders\SimpleCapture
Binaries (Elecard DirectShow Codec SDK root)\Bin\SimpleCapture.exe
(Elecard DirectShow Codec SDK root)\Bin\x64\SimpleCapture.exe

4.2.7.3 Features

Simple Capture implements the following features:

- Encoding video into AVC format
- Adjustment of base parameters of Elecard AVC Video Encoder using the application main window controls
- Access to all Elecard AVC Video Encoder settings (the **Advanced** button)
- Encoding audio into MPEG-1 or MPEG-2 Audio format
- Access to all Elecard MPEG Audio Encoder settings (the **Advanced** button in the **Audio Encoder** group)
- Multiplexing of the encoded streams into MPEG-2 format (TS/PS)
- Access to all of the capture device and crossbar filter settings
- Independent control of the filter graph Preview and Capture branches
- The built graph visualization using the Elecard Graph Viewer COM object

4.2.8 Low Delay Streamer

4.2.8.1 Description

Low Delay Streamer is a sample application that captures video from the specified screen area and audio from the specified PC audio device, transcodes the data into selected formats, multiplexes the encoded streams into MPEG-2 Transport Stream and broadcasts the stream to network clients. Due to the special adjustment of the Elecard components (so-called 'low delay' mode) the data processing speed is considerably increased.

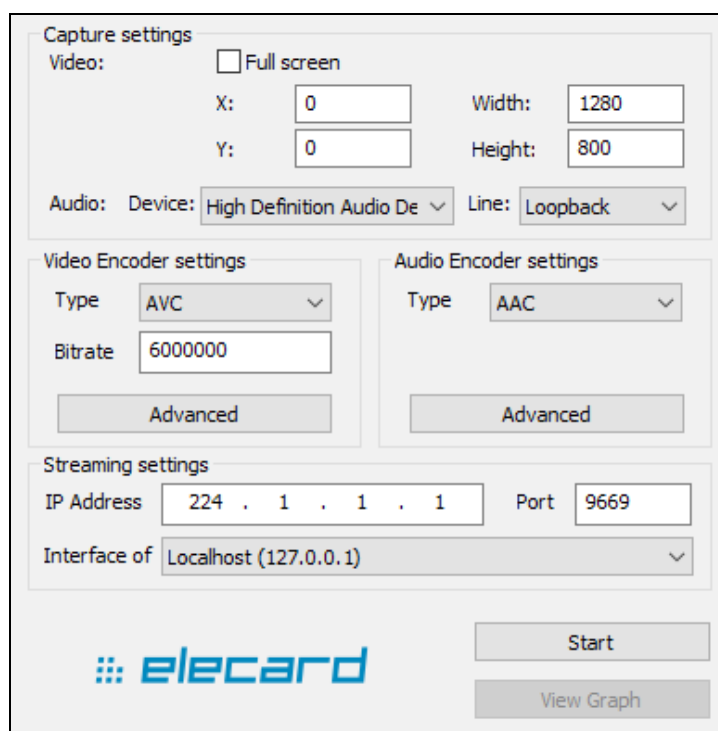


Figure 17. Low Delay Streamer GUI

Building of filter graph is performed after the **Start** button click. The **Stop** button click removes all components from the graph.

Video and audio capturing is performed with the Elecard Desktop Capture filter. The **Capture settings** group box of the application GUI represents the filter main features.

The **X** and **Y** parameters set the captured screen area top-left coordinates. The **Width** and **Height** parameters set the area dimensions. The **Full screen** option allows the full-screen capturing. The EDC_VideoCaptureRect parameter is used to pass the area coordinates to the Elecard Desktop Capture filter as the [left, top, right, bottom] array, where (left, top) and (right, bottom) are the top-left and bottom-right corner coordinates of the captured screen area, respectively.

To capture audio data, select audio device and input that provide the data.

To retrieve the list of available audio devices, use the *IParamConfig::EnumValidValues()* method for the EDC_AudioInputDevice parameter.

If it is not proposed to capture audio, set the 'not set' value instead of the audio board name. In this case the EDC_CaptureAudio parameter value is set to 0 and the Elecard Desktop Capture filter has only one output pin – video output pin.

For each audio board the list of available input lines is formed in the same way using EDC_AudioInputLine parameter.

The device list is compiled once in the *OnInitDialog()* function during the dialog initialization. While the input line list is updated every time, if a new device is selected. For Vista and later Windows versions there is the additional virtual 'Loopback' line as the hardware StereoMix (What You Hear) line replacement.

Low Delay Streamer allows selection of the data encoding formats (video – MPEG-2 or AVC, audio – MPEG or AAC) and adjustment of the encoded video bitrate. The **Advanced** button opens the **Settings** dialog for the corresponding encoder.

The main feature of video encoding in the Low Delay mode is absence of B-frames. The following values of the Elecard video encoder parameters are used for encoding streams without B-Frames:

Table 3. Elecard Video Encoders – Low Delay Parameter Values

Parameter GUID	Value	Description
Common		
EMC_GOP_MAX_LENGTH	15	Optimal GOP size for the Low Delay mode.
EMC_GOP_MAX_BCOUNT	0	Encoded stream does not contains B-frames.
EMC_GOP_ADAPTIVE	0	The parameter controls B-frame position in GOPs. It is disables as B-frames are absent in the encoded stream..
EMC_USE_SCD	0	The scene change detection (SCD) algorithm is disabled. It ensures the constant GOP size.
MPEG-2		
EM2VE_ZERO_INSTREAM	256	Number of zeros that are inserted into the stream after each encoded frame. It is used only in the Low Delay encoded streams that are intended for decoding with Elecard MPEG-2 Video Decoder.
EM2VE_WriteSh	1	Sequence header is contained in every GOP. It provides decoder with the stream information at any moment, if the decoding process is started.
EM2VE_WriteSec	1	The sequence end code is added to the encoded stream end. It provides decoder with the stream end additional information.
AVC		
EAVCENC_INITIAL_DELAY	30000	Sets the time interval that is used in decoding process for the initial input data buffering. The parameter influences on the encoding quality – less value causes lower quality level.

The additional parameters allowing acceleration of data processing (encoding, multiplexing, network

transferring) are adjusted in the *CbaseStreamer::LoadFilters()* function (the Low Delay Streamer application). The parameters are described in the corresponding component reference manuals.

For the data broadcasting, the broadcast IP address, port and network interface must be specified. These values are set via the corresponding parameters of the Elecard NWRenderer filter.

The network interface is an IP address of the network card that is used for the data broadcasting. The list of network interfaces is compiled in the *OnInitDialog()* function (with the standard *GetAdaptersInfo()* function) during the dialog initialization and contains the following values:

- not set – the interface is not set, the network card for broadcasting is selected automatically
- Localhost (127.0.0.1) – local broadcasting, the network card is not used
- available network cards – the selected card IP address is set as the network interface

The **View Graph** button displays the built filter graph.

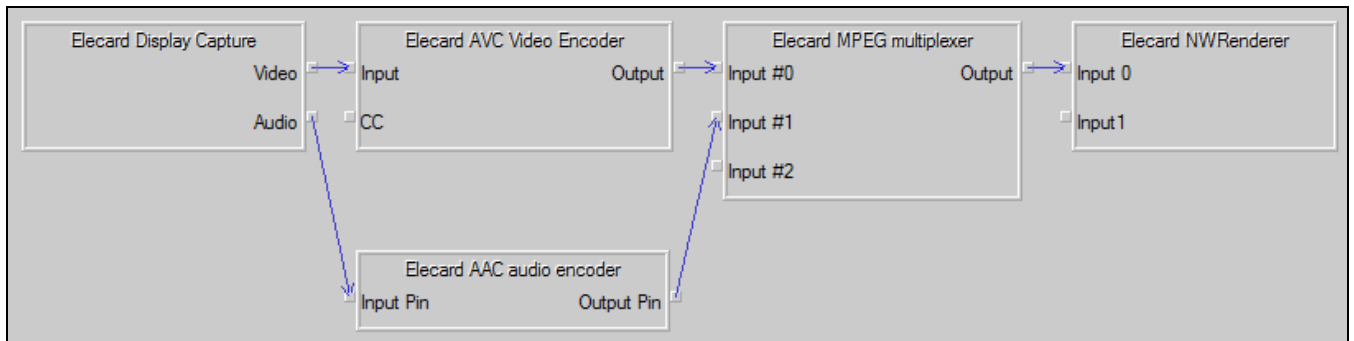


Figure 18. Low Delay Streamer Graph

4.2.8.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\LowDelay\LowDelayStreamer
Binaries (Elecard DirectShow Codec SDK root)\Bin\LowDelayStreamer.exe
 (Elecard DirectShow Codec SDK root)\Bin\x64\LowDelayStreamer.exe

4.2.9 Low Delay Client

4.2.9.1 Description

Low Delay Client is a sample application that demonstrates receiving and playback of video and audio data that is captured and broadcast with **Low Delay Streamer**.

The application user interface is very simple and contains only two buttons.

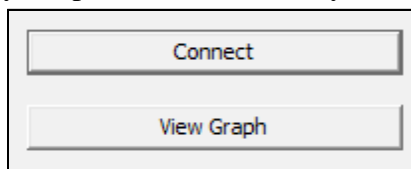


Figure 19. Low Delay Client GUI

To build the broadcast receiving filter graph, click the **Connect** button.

The Elecard NWSource-Plus filter is used as a source filter that receives data from network. The filter receives the list of available announcements. The first announcement with the *Desktop* session name is selected from the list. Then the filter is configured according to the requested and received SDP data. If the announcement with the *Desktop* session name is not received during 3 seconds, it is accepted as data absence and the message "Cannot build client graph. The 'Desktop' session is not available." is displayed. Otherwise, the source filter is connected to the Elecard MPEG Push Demultiplexer filter and the graph is set to the *Running* state for initialization of the demultiplexer output pins.

The demultiplexer initialization is completed, if the EMPGPDMX_INITIAL_PARSING_DONE parameter value is equal to 1. If the initialization is not completed during 30 seconds, it is accepted as data absence and the message "Cannot build client graph. The 'Desktop' session is not available." is displayed.

If the received stream contains both audio and video data, extra latency is added for the playback synchronization. That is the EMPGPDMX_LATENCY_VALUE demultiplexer parameter value is set to 2 (corresponds to 60 milliseconds). If only video is received, the synchronisation is not needed and the EMPGPDMX_LATENCY_VALUE parameter value is equal to 0.

During the further graph building the demultiplexer is connected to decoders and the decoders are connected to the corresponding rendering filters.

To reduce delay time to the minimal value, the decoders must be adjusted in the proper way.

The **View Graph** button displays the built filter graph.

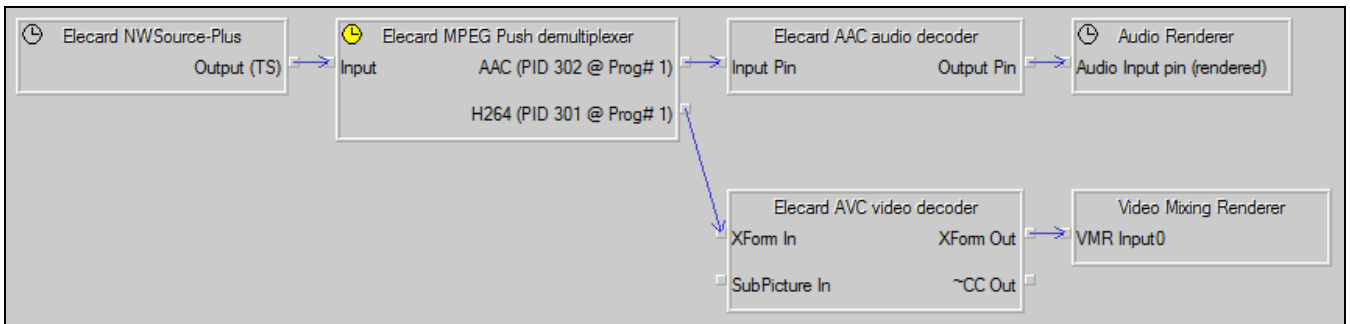


Figure 20. Low Delay Client Graph

The following parameter values provide minimal latency when decoding the streams encoded in the Low Delay mode:

Table 4. Elecard Video Decoders – Low Delay Parameter Values

Parameter GUID	Value	Description
Common		
EMC_HardwareAcceleration	0	Disables the DXVA hardware acceleration. It allows full decoding process control.
EMC_VMRMaintainAspectRatio	0	Disables the aspect ratio maintaining performed with the rendering filter. It provides correspondence of received and displayed frame sizes.
EMC_ErrorConcealment	0	The decoder filter sends all frames to the rendering filter (including frames decoded with errors). If this option is enabled, the corrupted frames are skipped during the rendering process. It causes the video jerking that seems as data receiving delay.
MPEG-2		
EMC_Reordering	0	Informs the decoder filter about absence of B-frames. It allows the decoder to output frames immediately after decoding in the input order.
AVC		
EAVCDEC_FullThreadLoad	0	Disables data waiting (is used for even loading of the decoding threads). If the option is enabled, the decoder waits data (up to several frames – depends on the number of CPU cores) before the decoding start. If the option is disabled, the decoding is started immediately when the data is available.
EAVCDEC_DecFrameBuffering	0	Disables waiting of the output queue filling. The decoder outputs a frame immediately after the frame is decoded.

4.2.9.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\LowDelay\LowDelayClient
Binaries (Elecard DirectShow Codec SDK root)\Bin\LowDelayClient.exe

(Elecard DirectShow Codec SDK root)\Bin\x64\LowDelayClient.exe

4.2.10 NWPlayer

4.2.10.1 Description

NWPlayer is a sample application that plays media streams from the network and allows the received media data saving onto a disk. It supports media files indexation using Elecard Sink Filter. The Elecard Module Configuration technology is used for the filters adjustment.

NWPlayer demonstrates work of the following filters:

- Elecard NWSource-Plus
- Elecard RTSP NetSource
- Elecard HLS Source
- Elecard Sink Filter

4.2.10.2 Using Elecard NWSource-Plus

The NWPlayer application builds the following graph to receive multicast/unicast multiplexed streams using the NWSource-Plus filter.

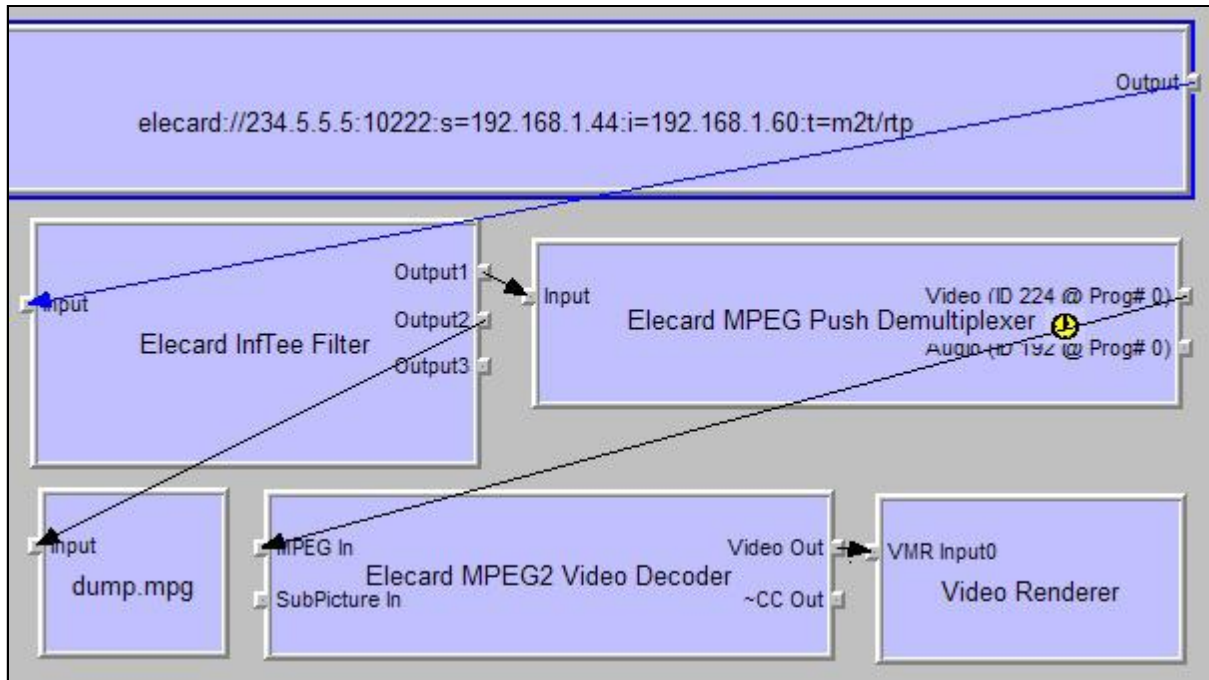


Figure 21. NWPlayer Filter Graph

Firstly, the application creates the instances of the NWSource-Plus, InfTee and Push Demultiplexer filters, adjusts NWSource-Plus via the **IModuleConfig** interface, connects the NWSource-Plus output pin with the InfTee filter and the InfTee output pin with the Push Demultiplexer filter. Then the application runs the graph and waits for creation of all of the Demultiplexer output pins.

Secondly, the application stops the graph, renders the Demultiplexer output video pin, creates the Sink Filter, connects it with the InfTee filter and runs the graph again.

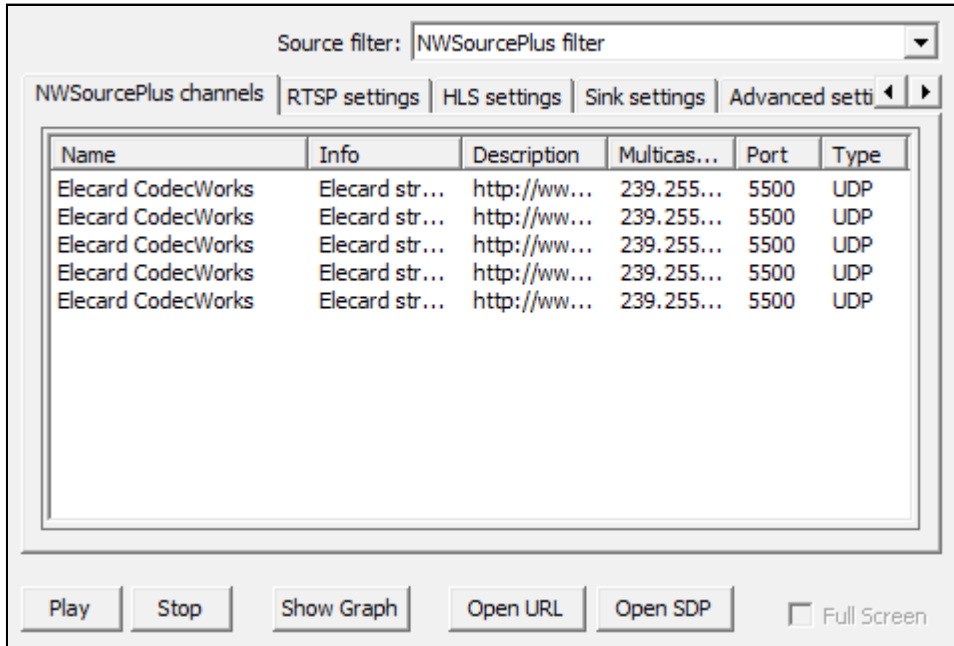


Figure 22. NWPlayer GUI – NWSourcePlus Channels Tab

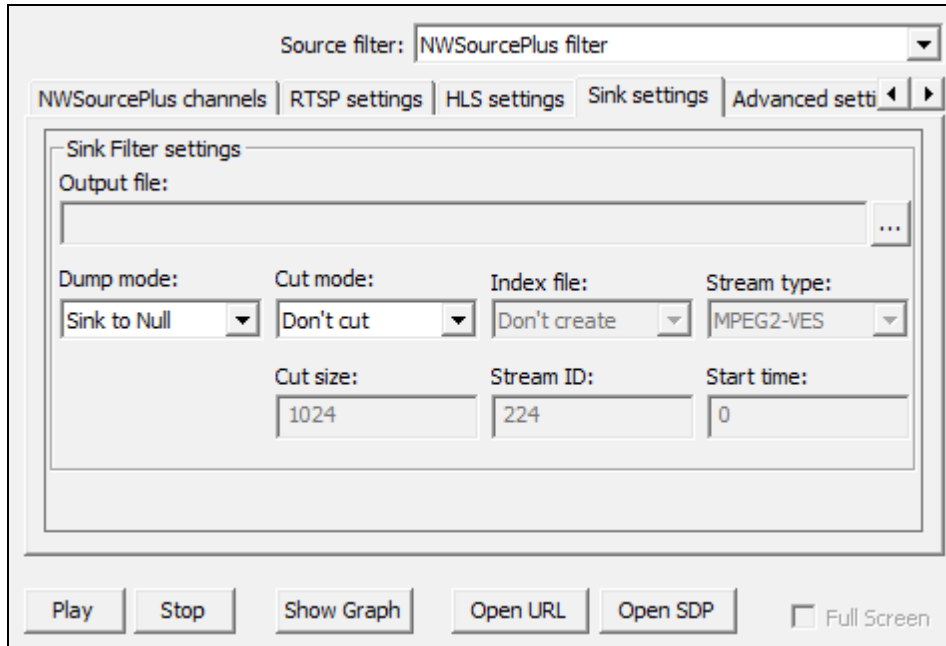


Figure 23. NWPlayer GUI – Sink Filter Settings Tab

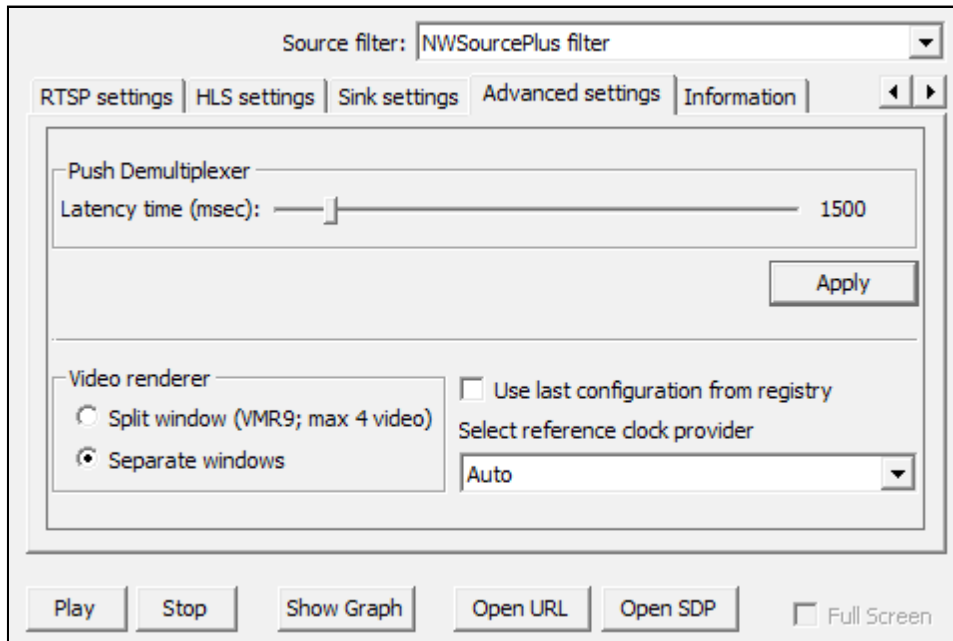


Figure 24. NWPlayer GUI – Advanced Settings Tab

The demultiplexer allows you to set delay time between the data receiving and playback using the **Latency time** property. To use the new value, click the **Apply** button.

The **Video renderer** section allows you to select the mode of multi-stream video presentation. If the **Separate window** mode is selected and the received stream contains more than one video stream, each video is displayed in its own window. If the **Split window** mode is selected, all video streams (no more than 4) are displayed in the single window.

4.2.10.3 Using Elecard RTSP NetSource

The NWPlayer sample application builds the following graph to receive multicast/unicast multiplexed streams using the RTSP protocol for exchange of commands between the application and server:

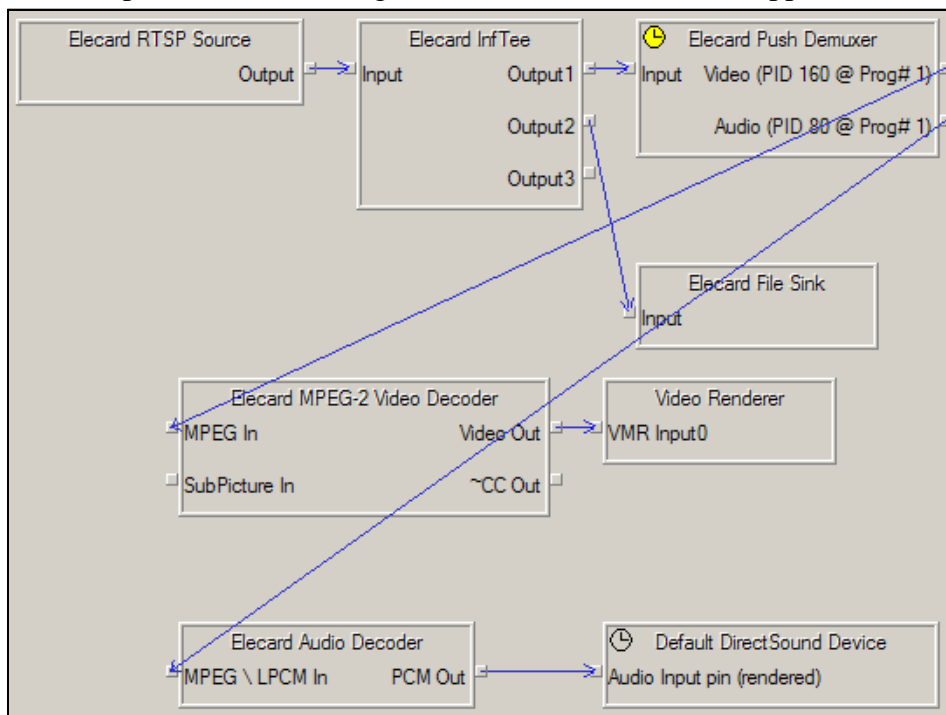


Figure 25. NWPlayer Filter Graph Using RTSP NetSource

Firstly, the application creates the instances of the RTSP NetSource, InfTee and Push Demultiplexer filters, adjusts RTSP NetSource via the **IModuleConfig** interface, and connects the RTSP NetSource output pin with the InfTee filter and the InfTee output pin with the Push Demultiplexer filter. Then the application runs the graph and waits for creation of all of the Demultiplexer output pins.

Secondly, the application stops the graph, renders the Demultiplexer output video pin, creates the Sink Filter, connects it with the InfTee filter and runs the graph again.

The Elecard RTSP NetSource filter requests the selected media file from VOD server with string of the following type:

```
elecard_rtsp://server_address:port[/data_path]
```

where:

server_address – server IP address

port – number of port used for RTSP connections (unsigned integer value from 1 to 65535; default value is 554)

data_path – media file for playback

For example: `elecard_rtsp://127.0.0.1:554/movie.mpg`

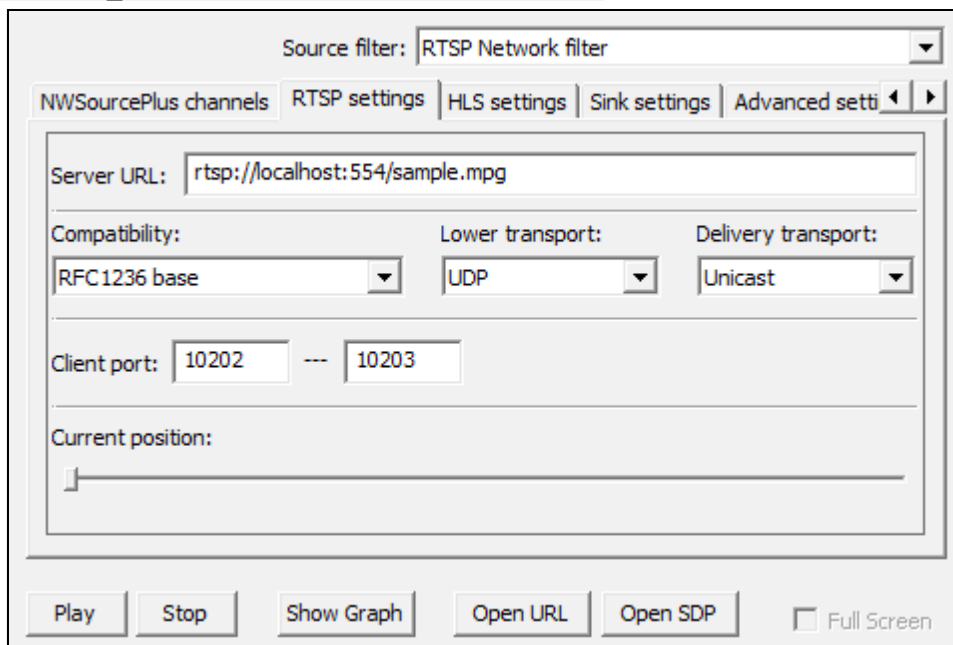


Figure 1. NWPlayer GUI - RTSP Settings Tab

4.2.10.4 Using Elecard HLS Source

The NWPlayer sample application builds the following graph to receive multimedia data using the HLS protocol:

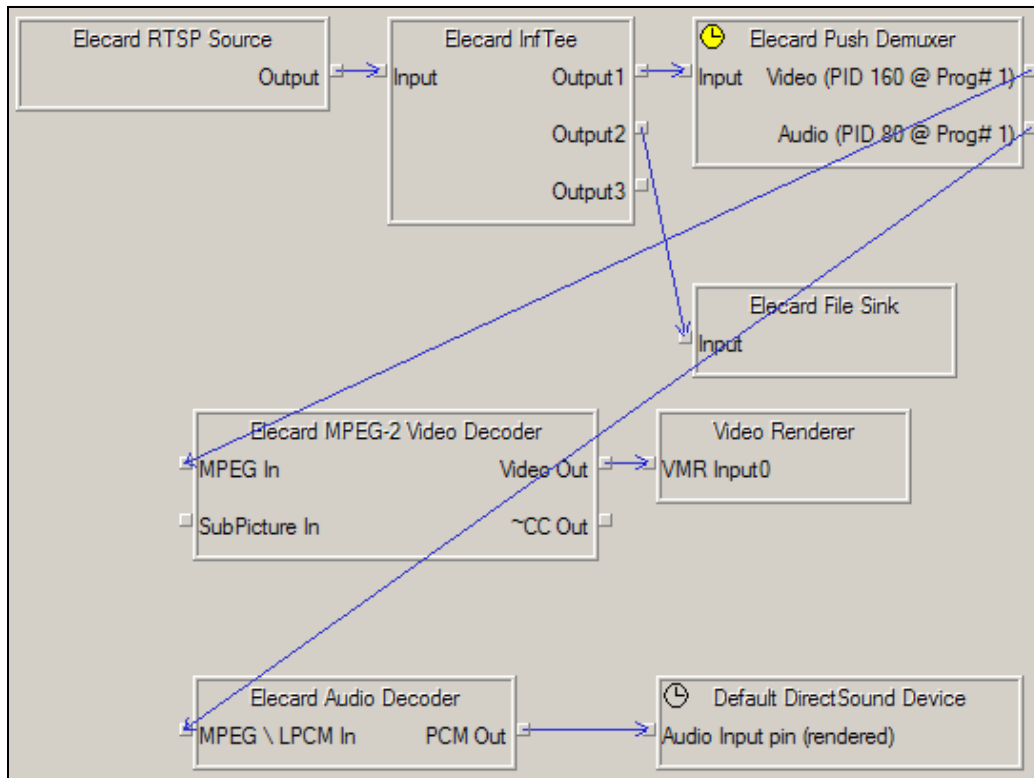


Figure 26. NWPlayer Filter Graph Using HLS Source

The graph is the same as with Elecard NWSrc-Plus and Elecard RTSP NetSource, but in HLS case the clock should be on the Audio Renderer filter (on the Default DirectSound Device as in the example). Otherwise, an asynchronous playback of audio and video streams is possible.

The Elecard HLS Source filter requests the media data from server with string of the following type:

`http://server_address/data_path`

where:

`server_address` – server IP address

`data_path` – path to m3u8 playlist for playback

For example:

`http://193.125.32.225/channel1.m3u8`

`http://hls-examples.com/playlist1.m3u8`

4.2.10.5 Running NWPlayer via Command Line

You can also run NWPlayer in various modes via command line.

UDP/RTP mode:

`NWPlayer.exe elecard://234.5.5.1:12222/rtp`

RTSP mode:

`NWPlayer.exe elecard_rtsp://192.168.1.109:554/Short.mpg`

HLS mode:

`NWPlayer.exe http://hls-examples.com/Playlist.m3u8`

4.2.10.6 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\Network\NWPlayer

Binaries (Elecard DirectShow Codec SDK root)\Bin\NWPlayer.exe

4.2.10.7 Features

The NWPlayer application supports the following features:

- Playback of media streams from the network (using NWSource-Plus, HLS Source or RTSP NetSource)
- Saving received media data onto a disk
- Indexation of the video data in MPEG-2 format (VES and Program Stream)

4.2.11 NWServer

4.2.11.1 Description

NWServer is a sample application that demonstrates the broadcasting media data to the network. It is capable of sending RTP, UDP and TCP packets and sends data session announcement via SAP (SDP) packets.

The NWServer sample application supports the following formats:

- MPEG-1 System Stream (ISO/IEC 11172-1)
- MPEG-2 Program Stream (ISO/IEC 13818-1)
- MPEG-2 Transport Stream (ISO/IEC 13818 –1)
- MP4

The following graphs are built in NWServer application:

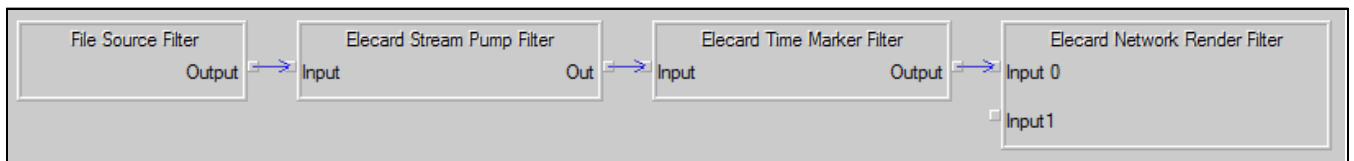


Figure 27. NWServer Filter Graph for MPEG-1/MPEG-2 Files

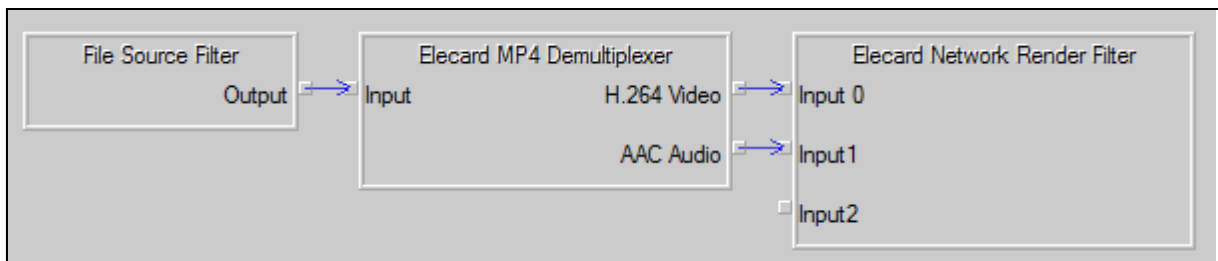


Figure 28. NWServer Filter Graph for MP4 Files

The following figures illustrate the NWServer GUI.

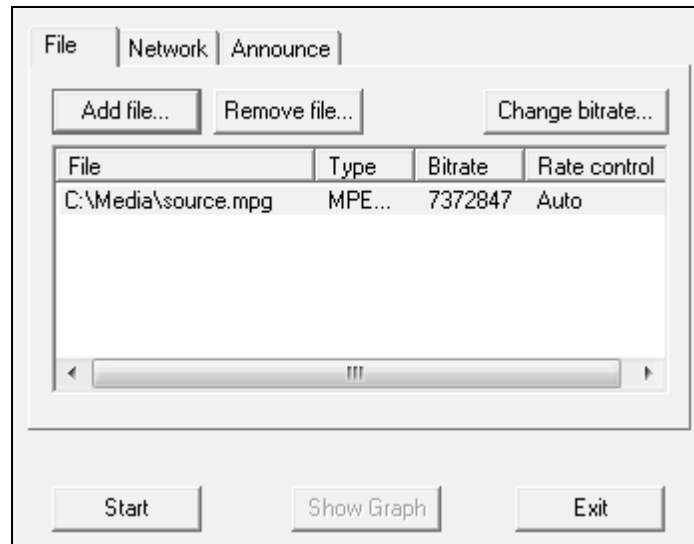


Figure 29. NWServer GUI - File Tab

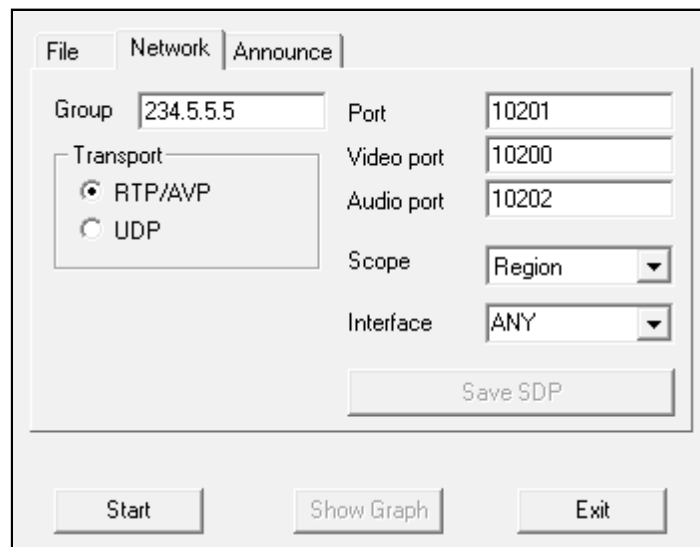


Figure 30. NWServer GUI - Network Tab

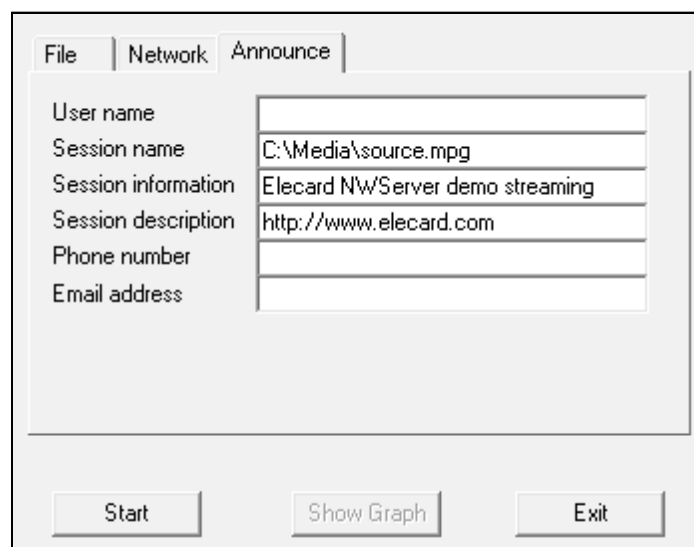


Figure 31. NWServer GUI - Announce Tab

4.2.11.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\Network\NWServer

Binaries (Elecard DirectShow Codec SDK root)\Bin\NWServer.exe

4.2.11.3 Features

The NWServer sample application performs the following tasks:

- Broadcasting (multicast or unicast) media streams over the Internet/Intranet
- Sending UDP, RTP and custom Multicast/Unicast streams
- Sending SAP announcement with SDP data
- Current bitrate monitoring
- Receiving media data from files
- Defining the list of files of the same type

4.2.12 NWServerTranscoder

4.2.12.1 Description

NWServerTranscoder is a sample application that demonstrates transcoding, multiplexing and broadcasting media data to the network. Broadcasting capabilities are the same as in the NWServer application.

The NWServerTranscoder sample application supports the following formats:

Input:

- Multiplexing formats – AVI, MPEG-2 TS/PS, MPEG-1 System, MP4
- Video formats – Uncompressed video, MPEG-1, MPEG-2, MPEG-4, AVC
- Audio formats – Uncompressed audio, MPEG, AAC

Output:

- Multiplexing formats – MPEG-2 TS/PS
- Video formats – MPEG-2, AVC
- Audio formats – MPEG, AAC

The following figures illustrate the NWServerTranscoder GUI.

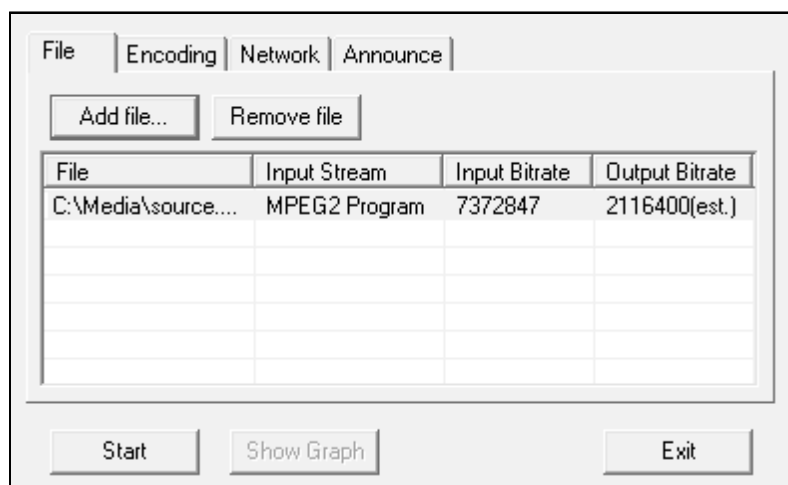


Figure 32. NWServerTranscoder GUI – File Tab

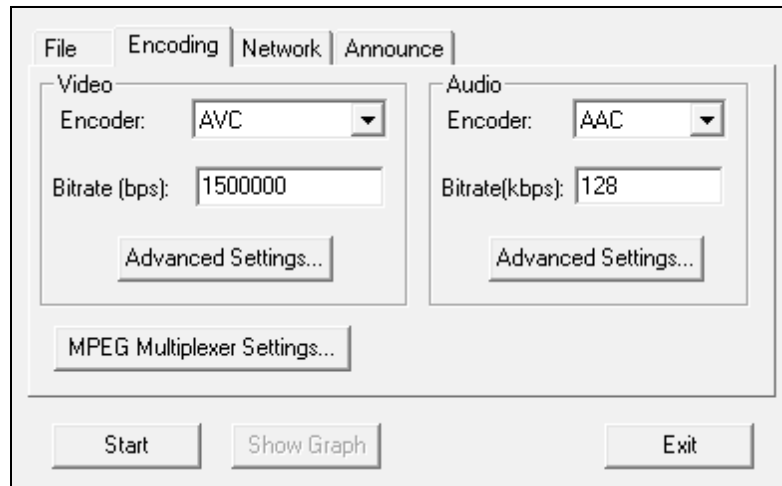


Figure 33. NWServerTranscoder GUI – EncodingTab

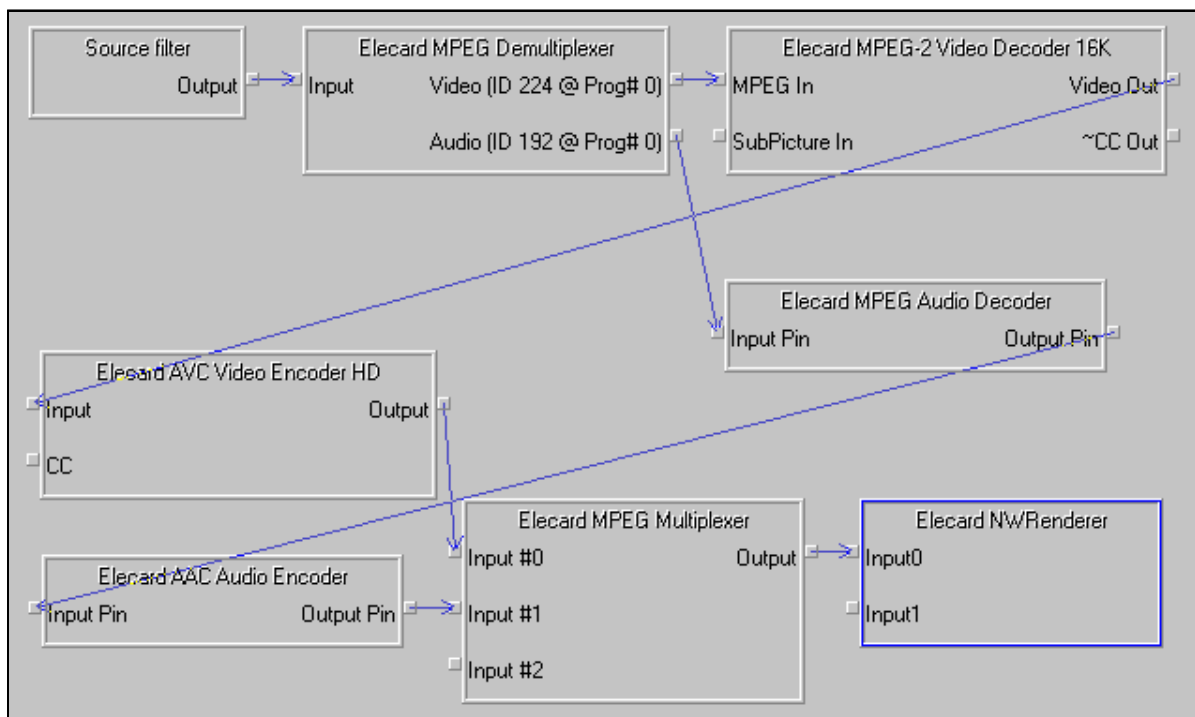


Figure 34. Filter Graph for File Transcoding and Broadcasting

The **Network** and **Announce** tabs are the same as in the NWServer sample application.

4.2.12.2 Path

Source (Elecard DirectShow Codec SDK root)\Sources\Samples\Network\NWServerTranscoder

Binaries (Elecard DirectShow Codec SDK root)\Bin\NWServerTranscoder.exe

4.2.12.3 Features

The NWServerTranscoder sample application performs the following tasks:

- Transcoding and multiplexing media streams
- Broadcasting (multicast or unicast) media streams over the Internet/Intranet
- Sending UDP, RTP and custom Multicast/Unicast streams
- Sending SAP announcement with SDP data
- Input and output bitrate monitoring