



**Elecard Video Transform SDK v.7.2**  
**Reference Manual**

## Notices

Elecard Video Transform SDK v.7.2 Reference Manual

First edition: April 2008

Date modified: February 3, 2009

For information, contact Elecard.

Tel: +7-3822-492-609; Fax: +7-3822-492-642

More information can be found at: [www.elecard.com](http://www.elecard.com)

For Technical Support, please contact the Elecard Technical Support Team: [tsup@elecard.net.ru](mailto:tsup@elecard.net.ru)

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 © 2007-2009 Elecard. All rights reserved.

# CONTENTS

<b>1. INTRODUCTION.....</b>	<b>4</b>
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 Base Classes.....	5
1.2.4 Sample Applications.....	5
1.3 REQUIREMENTS.....	6
1.3.1 Hardware Requirements.....	6
1.3.2 Software Requirements.....	6
1.4 TECHNICAL SUPPORT.....	6
<b>2. GETTING STARTED.....</b>	<b>7</b>
2.1 INTRODUCTION.....	7
2.2 INSTALLING ELECARD VIDEO TRANSFORM SDK.....	7
2.2.1 Uninstalling Elecard Video Transform SDK.....	7
2.3 DIRECTSHOW FILTERS ACTIVATION.....	7
2.3.1 Instance Activation.....	8
2.3.2 Adding a DirectShow Filter to the Graph without Registration.....	9
2.4 RUNNING ELECARD VIDEO TRANSFORM SDK SAMPLE APPLICATIONS.....	9
2.5 DESCRIBING ELECARD VIDEO TRANSFORM SDK FOLDER STRUCTURE.....	9
<b>3. DIRECTSHOW OVERVIEW.....</b>	<b>11</b>
3.1 INTRODUCTION.....	11
3.2 DIRECTSHOW MAIN TERMS AND DEFINITIONS.....	11
3.2.1 DirectShow Filter Types.....	11
3.2.2 Data Flow Principle.....	12
<b>4. BUILDING SIMPLE FILTER GRAPHS.....</b>	<b>13</b>
4.1 INTRODUCTION.....	13
4.2 SIMPLE PLAYER.....	13
4.2.1 Building Simple Player.....	13
<b>5. SAMPLE APPLICATIONS.....</b>	<b>16</b>
5.1 INTRODUCTION.....	16
5.1.1 Building sample applications.....	16
5.2 ELECARD VIDEO TRANSFORM SDK SAMPLE APPLICATIONS.....	16
5.2.1 2HD Video Scaler Player.....	17
5.2.2 Video Scaler Player.....	17
5.2.3 Deinterlace Player.....	18
5.2.4 Subtitle Player.....	19
5.2.5 Logo Player.....	20

# 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 Video Transform SDK. It includes information about the structure of the Elecard Video Transform SDK, provides a quick overview of the DirectShow fundamentals and features 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 Video Transform 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 Video Transform 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://www.microsoft.com>.

## 1.2 Preface

The Elecard Video Transform 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 Video Transform SDK package includes the following:

### 1.2.1 Documentation

Elecard Video Transform SDK documentation – consists of the following documents:

- Elecard Video Transform SDK Reference Manual (this document)

- Elecard Components Reference Manuals
- Elecard Base Classes Reference Manual
- 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 Video Transform SDK Components

Component	Description	File Name
<b>Elecard MPEG-2 Video Decoder</b>	DirectShow filter for MPEG-2 (ISO/IEC 13818-2) and MPEG-1 (ISO/IEC 11172-2) video decoding.	em2vd.ax
<b>Elecard MPEG-4 Video Decoder</b>	DirectShow filter for MPEG-4 (ISO/IEC 14496-2) video decoding.	em4vdec.ax
<b>Elecard AVC Video Decoder</b>	DirectShow filter for ISO/IEC 14496 part 10 AVC / ITU-T Recommendation H.264 video decoding.	eavcdec.ax
<b>Elecard MPEG Demultiplexer</b>	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
<b>Elecard MP4 Demultiplexer</b>	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.	emp4demux.ax
<b>Elecard Color Space Converter 2</b>	DirectShow filter that provides conversion between different formats of uncompressed video (for example, from RGB 24-bit to YUV 16-bit color).	ecsc2.ax
<b>Elecard Deinterlacer</b>	DirectShow filter that converts interlaced video into progressive scan format. The conversion is performed by using the edge-based line average algorithm (ELA) and the bicubic interpolation method.	edif.ax
<b>Elecard Logo Renderer</b>	DirectShow filter that allows rendering video overlaid with a preselected image. The image is overlaid on each video frame. The filter can also overlay the image by using the alpha channel.	elr.ax
<b>Elecard Subtitle Renderer</b>	Direct Show filter for the subtitle displaying. The subtitles are read from the XML file.	esr.ax
<b>Elecard Video Scaler</b>	DirectShow filter that allows high-quality video resizing, aspect ratio changing and working with vertical blanking interval (VBI) information.	evs.ax
<b>Elecard 2HD Video Scaler</b>	DirectShow filter that allows high-quality video resizing from any resolution to HD 720p, 720i, 1080p and 1080i formats.	e2hdvs.ax

## 1.2.3 Base Classes

**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.

## 1.2.4 Sample Applications

The Elecard Video Transform 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 Video Transform SDK Sample Applications

Sample Name	Description	File Name
<b>2HD Video Scaler Player</b>	Demonstrates use and adjustment of the Elecard 2HD Video Scaler filter. Represents basic player functionality (start, stop, fast forward, rewind, positioning, etc).	2HDVideoScalerPlayer.exe
<b>Video Scaler Player</b>	Demonstrates use and adjustment of the Elecard Video Scaler filter. Represents basic player functionality (start, stop, fast forward, rewind, positioning, etc).	VideoScalerPlayer.exe
<b>Deinterlace Player</b>	Demonstrates use and adjustment of the Elecard Deinterlacer filter. Represents basic player functionality (start, stop, fast forward, rewind, positioning, etc).	DeinterlacePlayer.exe
<b>Subtitle Player</b>	Demonstrates use and adjustment of the Elecard Subtitle Renderer filter. Represents basic player functionality (start, stop, fast forward, rewind, positioning, etc) and visualizes subtitles stored in <i>.XML</i> file.	SubtitlePlayer.exe
<b>Logo Player</b>	Overlays logo on video using the Elecard Logo Renderer filter. Represents basic player functionality (start, stop, fast forward, rewind, positioning, etc).	LogoPlayer.exe

## 1.3 Requirements

The Elecard Video Transform SDK has the following hardware and software requirements:

### 1.3.1 Hardware Requirements

Minimum hardware requirements:

- SSE-enhanced CPU (Intel® Pentium III, Celeron, AMD® Athlon, Opteron etc.)
- 128 MB RAM
- Any VGA card

### 1.3.2 Software Requirements

- Windows® 2000/XP/2003 Server
- Microsoft® DirectX® Media SDK/Microsoft® DirectX® SDK (version 8.1 or later)

---

*Note: Until April 2005, DirectShow was the part of DirectX. The last DirectShow patch for DirectX 9.0 is available in the DirectX 9.0 SDK Update (February 2005) Extras from <http://www.microsoft.com>. Then DirectShow was moved to the April 2005 Platform SDK. The DirectX SDK was, however, still required to build DirectShow samples. DirectShow was removed from the Platform SDK in 2006. Now Microsoft DirectShow 9.0 is available in the latest Windows SDK for Vista.*

---

## 1.4 Technical Support

For technical support contact the Elecard Technical Support Team:

[tsup@elecard.net.ru](mailto:tsup@elecard.net.ru)

## 2. Getting Started

---

### 2.1 Introduction

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

---

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

---

### 2.2 Installing Elecard Video Transform SDK

To install the Elecard Video Transform SDK:

1. Run the Elecard Video Transform SDK setup. To run, double click the executable file from the Elecard Video Transform SDK setup package.
2. The *Elecard Video Transform 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. Choose the type of installation: *Full* or *Custom*. Click **Next**. We recommend that you choose *Full*. Custom installation is for specific instances in which you know exactly which parts of the SDK you need.
6. Select the destination folder in which you want to install the Elecard Video Transform SDK. Click **Next**.
7. Select the program group in which you want the Elecard Video Transform SDK to be located. Click **Next**.
8. 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 Video Transform SDK has been successfully installed*” will appear and the SDK is ready to use.

#### 2.2.1 Uninstalling Elecard Video Transform SDK

To uninstall the Elecard Video Transform SDK, click *Start*→*Programs*→*Elecard*→*Elecard Video Transform SDK xx*→*Uninstall Elecard Video Transform SDK* (xx – the SDK version number).

Follow the on-screen instructions to complete removal of the SDK.

### 2.3 DirectShow Filters Activation

Most of the decoder, encoder, and transform (video and audio) filters included into the SDK package have a

copy-protection mechanism: without activation these filters operate in an evaluation mode (e.g. overlay logo on the video) and have 30 days evaluation period limitation. After activation filters operate in a demo mode (e.g. still overlay logo on the video, but without time limitation and restrictions imposed on an expired mode). When the development is finished, the OEM pack with the components used in the product must be ordered (licenced) from Elecard.

There are two ways to perform activation:

1. Activation via the the *Elecard Registrator* program using a special activation number. In this case every end-user should start the *Registrator* program and type the unique activation number. This activation method is intended for end-users.
2. "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.

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=xxxxxxxxxxxxxxxxxxxx"` 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 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 (described below) 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.

### 2.3.1 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.

### 2.3.2 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 shows how you can create a filter using the file path.

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

where *m\_EncoderPath* is path of the encoder binary file.

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

## 2.4 Running Elecard Video Transform SDK Sample Applications

To run the Elecard Video Transform SDK sample applications:

Click *Start*→*Programs*→*Elecard*→*Elecard Video Transform SDK xx*→*Bin* and select application (xx – the SDK version number).

## 2.5 Describing Elecard Video Transform SDK Folder Structure

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

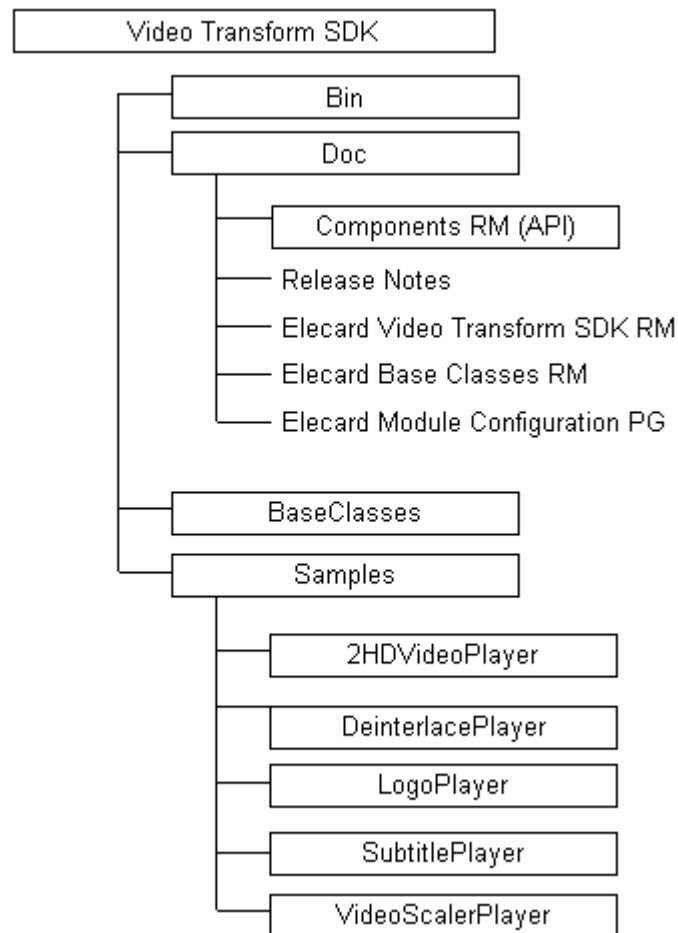
---

**Note:** After installing the Elecard Video Transform SDK, the SDK Components (filters and DLLs) will be installed on your PC in the “*Program Files\Elecard\Elecard Video Transform SDK\Bin*” folder.

---

The SDK folder has the following structure:

Figure 1. Elecard Video Transform SDK Folder Structure



The SDK folder contains:

1. **Bin** – contains the SDK binary files (DS filters, engine DLLs, samples executable binaries).
2. **Doc** – includes all SDK-related documentation.
3. **Base Classes** – includes the source codes of the Elecard Video Transform SDK base class libraries.
4. **Samples** – contains the source codes of the Elecard Video Transform SDK sample applications.

## 3. DirectShow Overview

---

### 3.1 Introduction

The following section provides an overview of the DirectShow components, including filter types and data flow graphs. For detailed graph building instruction, see the Building Simple Filter Graphs section.

### 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 Video Transform SDK filters are installed in the “*Program Files\Elecard\Elecard Video Transform SDK\Bin\*” 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.

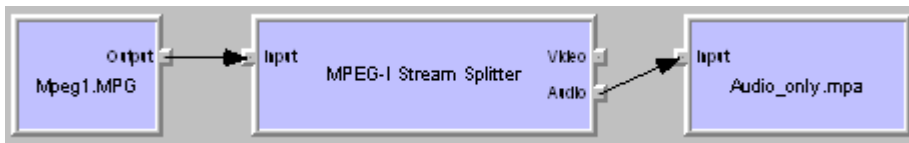
### 3.2.2 Data Flow Principle

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:

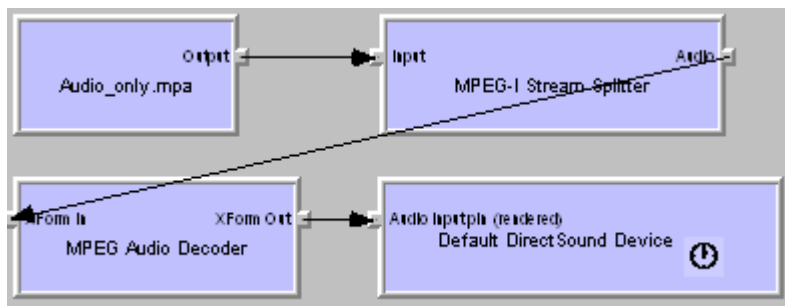
- **Async File Source** – gets data from the MPEG-1 file.
- **MPEG-1 Splitter** – splits MPEG-1 streams into MPEG-1 video and MPEG-1 audio elementary streams.
- **File Dump Filter** – writes compressed audio to the hard drive.

Figure 2. 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 3. Dumped Audio File Playback Filter Graph



## 4. Building Simple Filter Graphs

### 4.1 Introduction

This section provides instructions for building simple graphs using Elecard Video Transform SDK filters.

### 4.2 Simple Player

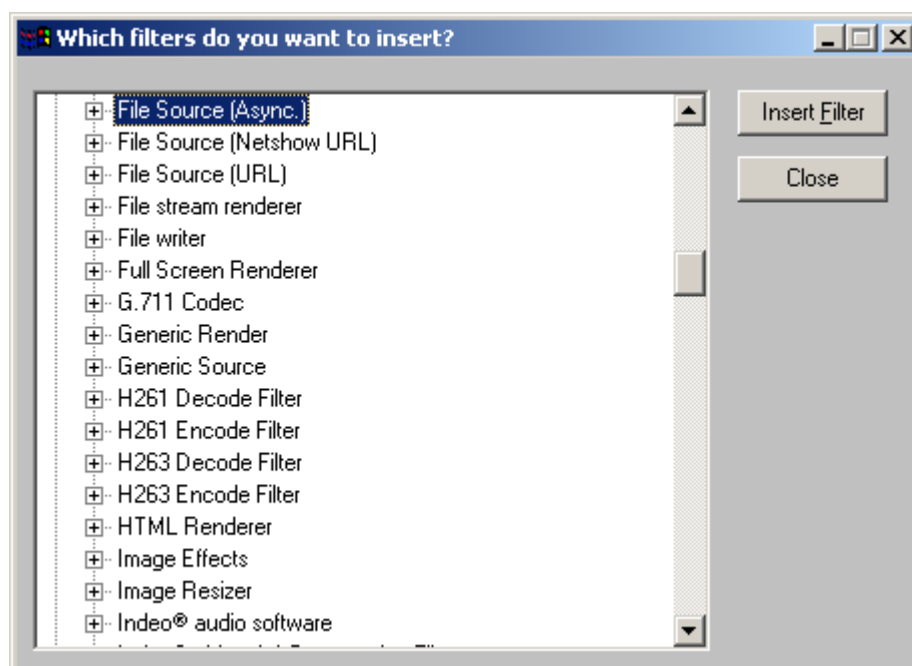
This Simple Player example demonstrates the building of the DirectShow graph for playback of an MPEG-1/ MPEG-2 media files using the standard File Source filter and other Elecard developed filters such as: Elecard MPEG-2 Video Decoder.

#### 4.2.1 Building Simple Player

To build the Simple Player:

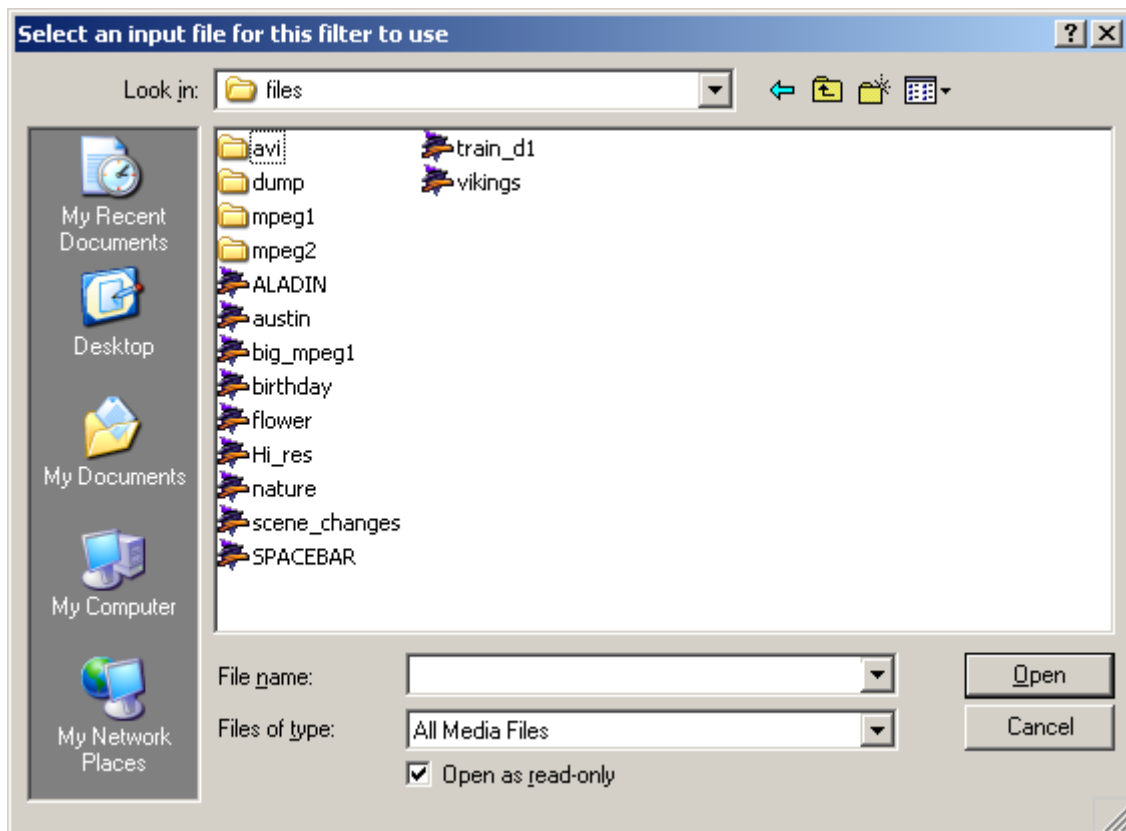
1. Start GraphEdit from the installed DirectX SDK.  
(*Start→Programs→Microsoft DirectX SDK→DirectX Utilities→ GraphEdit*).
2. On the **Graph** menu, click **Insert Filters**.
3. From the “Which filters do you want to insert?” window, open DirectShow Filters, choose **File Source (Async.)** and click **Insert Filter**.

Figure 4. GraphEdit ‘Insert Filters’ Dialog



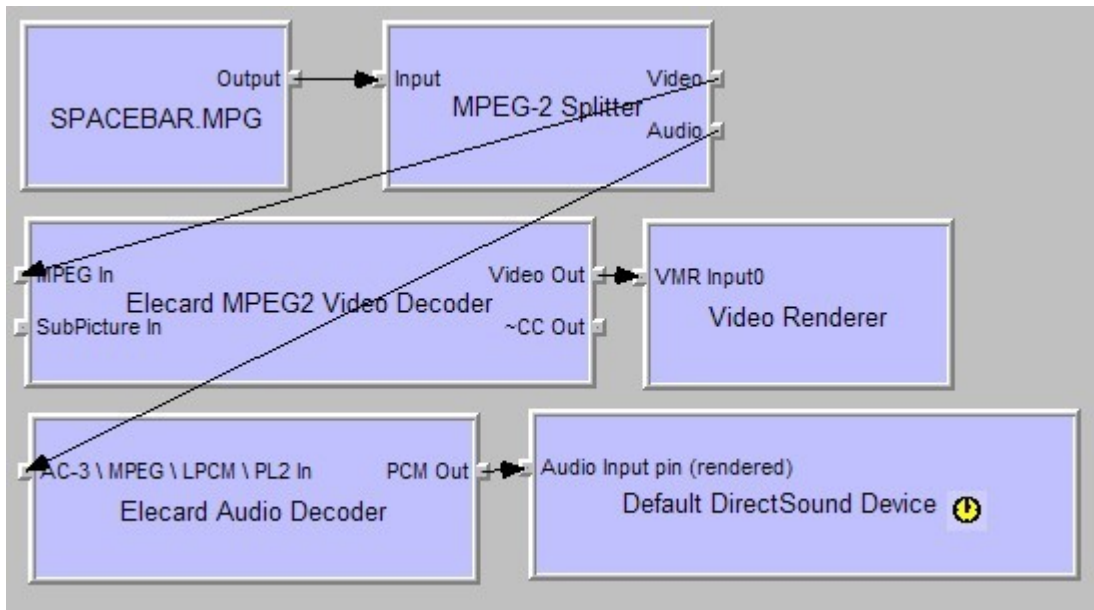
4. The “Select an input file...” window will appear. Choose the MPEG file you want to render.

Figure 5. 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 (or MS MPEG-2 Splitter), Elecard MPEG-2 Video 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.
9. The following graph will appear:

Figure 6. Simple Player Filter Graph



10. To start playback, on the **Graph** menu, click **Play**.

*Note: For additional information on working with GraphEdit see the GraphEdit help.*

## 5. Sample Applications

---

### 5.1 Introduction

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

#### 5.1.1 Building sample applications

---

**Note:** You need to install the Microsoft® DirectX® SDK **before** building samples included in the Elecard Video Transform SDK.

---

To build the sample applications it is necessary to perform the following actions:

1. Build and link the DirectShow BaseClasses library.

In DirectX 8.1 the sources of the Base Classes are stored in the directory: *(SDK root)\Samples\Multimedia\DirectShow\BaseClasses*

In DirectX 9.0 the sources of the Base Classes are stored in the directory: *(SDK root)\Samples\C++\DirectShow\BaseClasses*

As the result the *strmbase.lib* and *strmbasd.lib* files will be created.

2. Add paths to BaseClasses header files, *strmbase.lib* and *strmbasd.lib* to Visual C++ settings. The alternative way is to put *strmbase.lib* and *strmbasd.lib* to the directory *(SDK root)\lib\*, the path of which also must be set in Visual C++ settings.
3. To build samples from Elecard Video Transform SDK open the Workspace file (*“Elecard Video Transform SDK root)\Samples\VideoTransformSamples.sln”*). In this Workspace there are all the projects of the SDK samples and the project of Elecard BaseClasses. All the necessary dependences among the projects are already set; so on building any sample all the necessary libraries of BaseClasses will be automatically built and linked. The relative paths to the Elecard BaseClasses are included into the MSVC project files as "Additional include directories". The paths to the libraries are also set in the MSVC project files as "Additional library path". If you want to replace a sample project, do not forget to change corresponding paths or replace the project together with BaseClasses keeping the structure of directories.
4. To build the any another SDK samples open the corresponding Workspace file and act in the same way as described in the previous item.

### 5.2 Elecard Video Transform SDK Sample Applications

The following describes the Player SDK Samples, including:

- 2HD Video Scaler Player
- Video Scaler Player
- Deinterlace Player

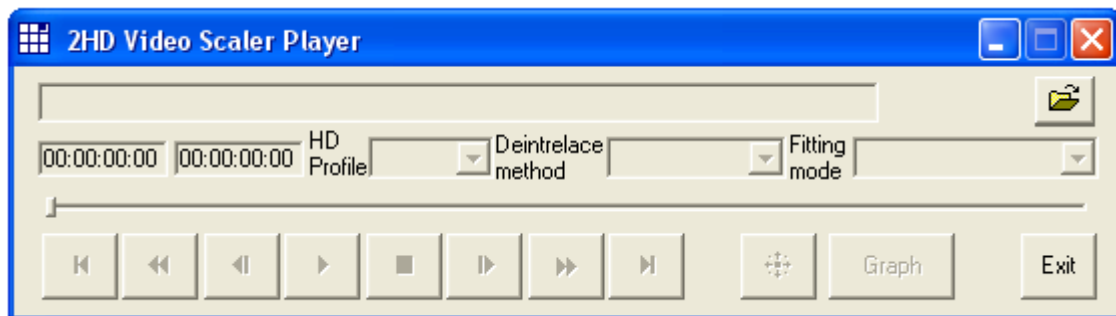
- Subtitle Player
- Logo Player

## 5.2.1 2HD Video Scaler Player

### 5.2.1.1 Description

**2HD Video Scaler Player** is a sample application that demonstrates use and adjustment of the Elecard 2HD Video Scaler filter. The application represents basic player functionality (start, stop, fast forward, rewind, positioning, etc). **IModuleConfig** interface is used for the filter parameters configuration.

Figure 7. Video Scaler Player GUI



### 5.2.1.2 Path

**Source** (Elecard Video Transform SDK root)\Samples\2HDVideoScalerPlayer  
**Binaries** (Elecard Video Transform SDK root)\Bin\2HDVideoScalerPlayer.exe

### 5.2.1.3 Features

2HD Video Scaler Player implements the following features:

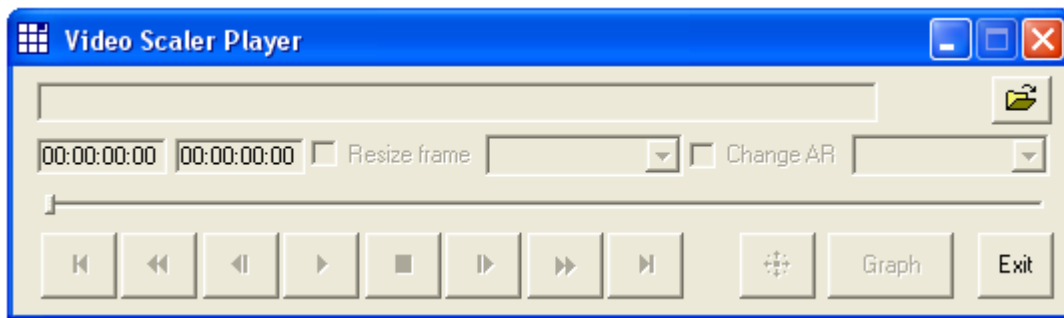
- Media file playback
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Video resizing in accordance with 4 predefined HDTV profiles (720i, 720p, 1080i, 1080p)
- Video deinterlacing in accordance with selected method
- Video fitting in accordance with selected mode (*Fit to destination* – video resizing without taking original aspect ratio into consideration. *Based on vertical/horizontal size* – video resizing with regard to specified vertical/horizontal size and original aspect ratio.)
- Possibility to view the built graph using the Elecard DSGViewer COM object

## 5.2.2 Video Scaler Player

### 5.2.2.1 Description

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

Figure 8. Video Scaler Player GUI



### 5.2.2.2 Path

**Source** (Elecard Video Transform SDK root)\Samples\VideoScalerPlayer

**Binaries** (Elecard Video Transform SDK root)\Bin\VideoScalerPlayer.exe

### 5.2.2.3 Features

Video Scaler Player implements the following features:

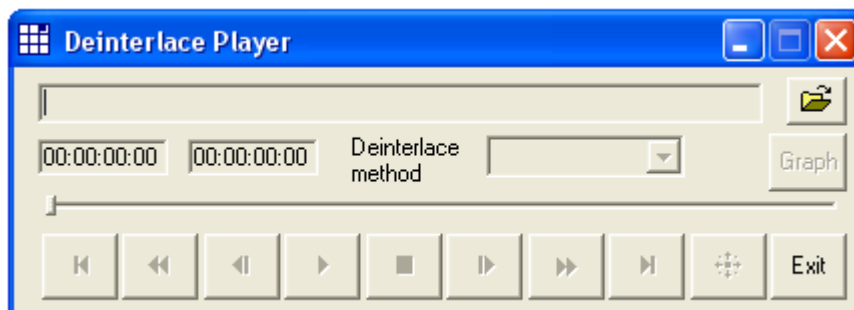
- Media file playback
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Video resizing into the most popular video resolutions (640x480, 720x480, 720x576, 1280x720, etc)
- Aspect ratio adjustment (standard values are available – 1:1; 4:3; 16:9; 2:1, etc)
- Possibility to view the built graph using the Elecard DSGViewer COM object

## 5.2.3 Deinterlace Player

### 5.2.3.1 Description

**Deinterlace Player** is a sample application that demonstrates use and adjustment of the Elecard Deinterlacer filter. The application represents basic player functionality (**Start, stop, fast forward, rewind, positioning**, etc). **IModuleConfig** interface is used for the filter parameters configuration.

Figure 9. Deinterlace Player GUI



### 5.2.3.2 Path

**Source** (Elecard Video Transform SDK root)\Samples\DeinterlacePlayer

**Binaries** (Elecard Video Transform SDK root)\Bin\DeinterlacePlayer.exe

### 5.2.3.3 Features

Deinterlace Player implements the following features:

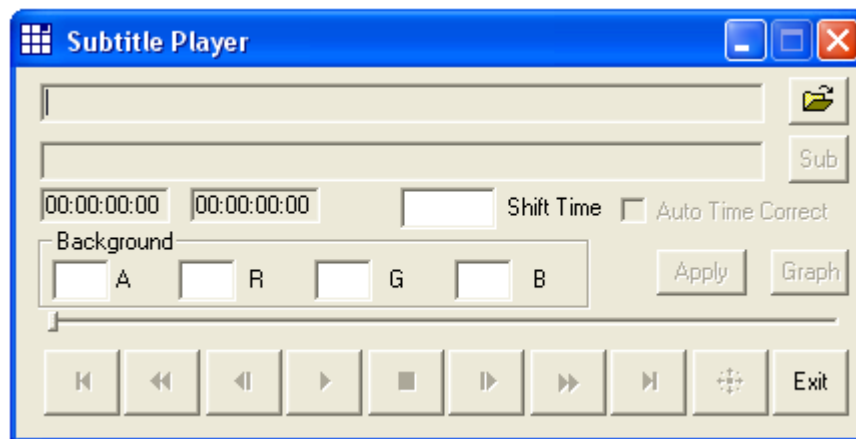
- Media file playback
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Video deinterlacing in accordance with selected method ((High speed or High quality)
- Possibility to view the built graph using the Elecard DSGViewer COM object

## 5.2.4 Subtitle Player

### 5.2.4.1 Description

**Subtitle Player** is a sample application that demonstrates use and adjustment of the Elecard Subtitle Renderer filter. The application represents basic player functionality (**Start, stop, fast forward, rewind, positioning,** etc) and visualizes subtitles stored in *.XML* file. **IModuleConfig** interface is used for the filter parameters configuration.

Figure 10. Subtitle Player GUI



### 5.2.4.2 Path

<b>Source</b>	(Elecard Video Transform SDK root)\Samples\SubtitlePlayer
<b>Binaries</b>	(Elecard Video Transform SDK root)\Bin\SubtitlePlayer.exe
<b>Subtitles Example</b>	(Elecard Video Transform SDK root)\Bin\FACE_CACHEE.XML

### 5.2.4.3 Features

Subtitle Player implements the following features:

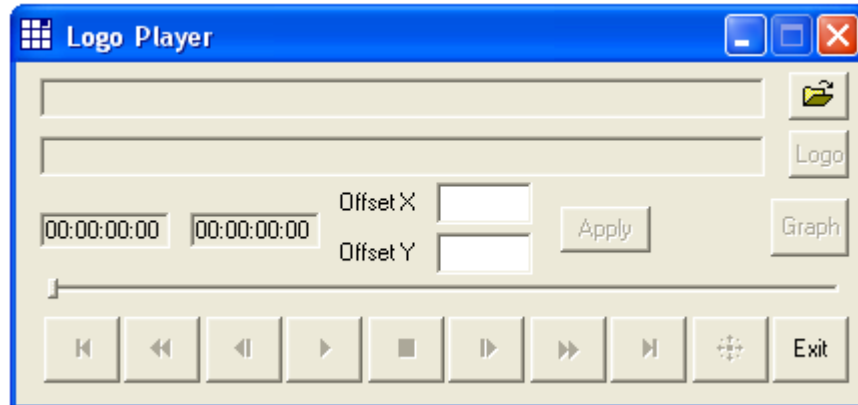
- Media file playback
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Automatic time correction
- Shifting time (seconds) if necessary
- Adjustment of the subtitles background color (R, G, B 0-255) and opacity (A 0-255) within limits 0-255
- Possibility to view the built graph using the Elecard DSGViewer COM object

## 5.2.5 Logo Player

### 5.2.5.1 Description

**Logo Player** is a sample application that overlays logo on video using the Elecard Logo Renderer filter. The application represents basic player functionality (start, stop, fast forward, rewind, positioning, etc) and visualizes subtitles stored in *.XML* file. **IModuleConfig** interface is used for the filter parameters configuration.

Figure 11. Logo Player GUI



### 5.2.5.2 Path

<b>Source</b>	(Elecard Video Transform SDK root)\Samples\LogoPlayer
<b>Binaries</b>	(Elecard Video Transform SDK root)\Bin\LogoPlayer.exe
<b>Logo Example</b>	(Elecard Video Transform SDK root)\Bin\LogoElecard.bmp

### 5.2.5.3 Features

Logo Player implements the following features:

- Media file playback
- Basic player functionality (start, stop, fast forward, rewind, positioning, etc)
- Adjustment of the logo position in the video window
- Possibility to view the built graph using the Elecard DSGViewer COM object