SolveigMM Video Editing SDK

Developer Reference Manual

SDK Version: 3.0 FULL
First edition: February 27, 2007
Date modified: September 12, 2013

Solveig Multimedia,
Razvitiya ave 3, 634055,
Tomsk, Russian Federation

www.solveigmm.com
Contents

SolveigMM Video Editing SDK Notices ................................................................. 6
Release Notes ....................................................................................................... 7
Product Description ............................................................................................. 7
New features in SolveigMM Video Editing SDK 3.0 .............................................. 9
Components ......................................................................................................... 11
Sample applications ............................................................................................ 14
  C++ Samples ....................................................................................................... 14
    C++ Trimming Samples .................................................................................... 14
    C++ Multiplexing Samples ............................................................................. 15
    C++ Joining Samples ...................................................................................... 16
    Other C++ Samples ....................................................................................... 18
  C# Samples ....................................................................................................... 18
    C# Trimming Samples: .................................................................................... 18
    C# Joining Samples: ...................................................................................... 19
    Other C# Samples: ......................................................................................... 21
Delphi Samples ..................................................................................................... 21
  Delphi Trimming Samples: ................................................................................ 21
  Delphi Joining Samples: .................................................................................... 22
  VB.60 Samples .................................................................................................. 23
  VB.Net Samples ............................................................................................... 23
    VB.Net Trimming Samples ............................................................................ 23
    VB.Net Joining Samples ................................................................................. 23
System requirements ........................................................................................... 26
User Guide ........................................................................................................... 27
SDK folder structure ............................................................................................ 27
Installing SDK ....................................................................................................... 27
Uninstalling SDK ................................................................................................. 27
SDK Sample Applications ..................................................................................... 28
  C++ Samples ..................................................................................................... 28
    C++ Trimming Samples .................................................................................. 28
      SMM Trim C++: AVI, ASF ........................................................................... 28
      SMM Trim C++: MP3, WMA, WAV ............................................................... 28
      SMM Trim C++: MPEG1, MPEG2 ............................................................... 29
      SMM Trim C++: AVCHD, MP4, MOV, FLV, MKV, WEBM ......................... 29
    C++ Multiplexing Samples ............................................................................. 30
      SMM Mux C++: ASF ................................................................................... 31
      SMM Mux C++: MKV, MP4, MOV, WEBM .................................................. 32
      SMM Mux C++: FLV ................................................................................... 32
    C++ Joining Samples ...................................................................................... 33
      SMM Join C++: AVCHD, AVI, MKV, MOV, MP3, MP4, FLV, MPEG1, MPEG2, WAV, WMA, WEBM .......................................................... 33
      Other C++ Samples ...................................................................................... 33
  SolveigMM Batch Splitter Sample Application .................................................. 33
  SolveigMM ASF Markers ................................................................................... 33
  SolveigMM AVI Trimmer Console Utility .......................................................... 33
  SolveigMM MP4 to ASF Playready ..................................................................... 33
  SolveigMM Video Splitter .................................................................................. 34
  SolveigMM Screenshot Grabber ....................................................................... 34
  C# Samples ....................................................................................................... 35
    SMM Trim C#: AVCHD, AVI, MKV, MOV, MP3, MP4, FLV, MPEG1, MPEG2, WAV, WMA, WEBM ................................................................. 35
    C# Joining Samples ....................................................................................... 36
    Other C# Samples ......................................................................................... 36
SolveigMM Video Editing SDK Reference Manual

SolveigMM Video Editing Engine

Editing Engine enumerations

- SMM_OutputTypes
- SMM_TaskType
- SMM_TrimmAccuracyType
- SMM_TrimFlags

Editing Engine structures

- SMM_GROUP_TASK
- SMM_TRACK_INFO
- SMM_CLIP_INFO
- TrimInfoList
- TSTrimInfo
- FILE_VERSIONS_2
- VERSION_INFO_2
- SMM_ASF_MARKER

Editing Engine Interfaces

- IModuleConfig interface
- ITrimmerObjControl interface
  - Start
  - Stop
- ITrimmerObjControlCB interface
  - OnStart
  - OnStop
  - OnError

SolveigMM Video Editing Engine Parameters

- SMAT_InputFile
- SMAT_OutputFile
- SMATFlags
- SMAT_ResetFlags
- SMAT_TrimList
- SMAT_Progress
- SMAT_AudioStreamNumber
- SMAT_VideoStreamNumber
- SMAT_Callback
- SMAT_GetVersions2
- SMAT_CurrentSize
- SMAT_BatchFileName
- SMAT_CurrentBatchTask
- SMAT_OutputType
- SMAT_BatchTasksNum
- SMAT_TaskType
Solveig Multimedia

SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_TrackInfo</td>
<td>58</td>
</tr>
<tr>
<td>SMAT_MPEG2IndexFileName</td>
<td>58</td>
</tr>
<tr>
<td>SMAT_SilentPath</td>
<td>58</td>
</tr>
<tr>
<td>SMAT_MPEG2CodecsSet</td>
<td>59</td>
</tr>
<tr>
<td>SMAT_CodecsSet</td>
<td>59</td>
</tr>
<tr>
<td>SMAT_GetFileType</td>
<td>59</td>
</tr>
<tr>
<td>SMAT_VidateFiles</td>
<td>60</td>
</tr>
<tr>
<td>Required steps before start joining</td>
<td>61</td>
</tr>
<tr>
<td>SMAT_ASFMarker</td>
<td>62</td>
</tr>
<tr>
<td>SMAT_ASFMarkersCount</td>
<td>62</td>
</tr>
<tr>
<td>SMAT_ASFMarkerRemove</td>
<td>62</td>
</tr>
<tr>
<td>SolveigMM BMP to AVI</td>
<td>63</td>
</tr>
<tr>
<td>BmpToAvi.dll</td>
<td>63</td>
</tr>
<tr>
<td>Requirements</td>
<td>63</td>
</tr>
<tr>
<td>Features</td>
<td>63</td>
</tr>
<tr>
<td>Usage</td>
<td>64</td>
</tr>
<tr>
<td>SolveigMM BMP Push Source Filter</td>
<td>64</td>
</tr>
<tr>
<td>Requirements</td>
<td>64</td>
</tr>
<tr>
<td>Features</td>
<td>65</td>
</tr>
<tr>
<td>Usage</td>
<td>65</td>
</tr>
<tr>
<td>SolveigMM Media Joiner Filter</td>
<td>68</td>
</tr>
<tr>
<td>Requirements</td>
<td>68</td>
</tr>
<tr>
<td>Features</td>
<td>68</td>
</tr>
<tr>
<td>Usage</td>
<td>69</td>
</tr>
<tr>
<td>SolveigMM MKV Muxer Filter</td>
<td>79</td>
</tr>
<tr>
<td>Features</td>
<td>79</td>
</tr>
<tr>
<td>Usage</td>
<td>79</td>
</tr>
<tr>
<td>SolveigMM MP4 Muxer filter</td>
<td>84</td>
</tr>
<tr>
<td>Features</td>
<td>85</td>
</tr>
<tr>
<td>Usage</td>
<td>85</td>
</tr>
<tr>
<td>SolveigMM MP4 Splitter filter</td>
<td>86</td>
</tr>
<tr>
<td>Features</td>
<td>86</td>
</tr>
<tr>
<td>Usage</td>
<td>87</td>
</tr>
<tr>
<td>SolveigMM FLV Muxer filter</td>
<td>88</td>
</tr>
<tr>
<td>Features</td>
<td>88</td>
</tr>
<tr>
<td>Usage</td>
<td>88</td>
</tr>
<tr>
<td>SolveigMM FLV Splitter filter</td>
<td>89</td>
</tr>
<tr>
<td>Features</td>
<td>89</td>
</tr>
<tr>
<td>Usage</td>
<td>90</td>
</tr>
<tr>
<td>SolveigMM ASF Multiplexer filter</td>
<td>91</td>
</tr>
<tr>
<td>Features</td>
<td>91</td>
</tr>
<tr>
<td>Usage</td>
<td>91</td>
</tr>
<tr>
<td>SolveigMM-Elecard MPEG2 Trimmer filter</td>
<td>90</td>
</tr>
<tr>
<td>Requirements</td>
<td>99</td>
</tr>
<tr>
<td>Features</td>
<td>99</td>
</tr>
<tr>
<td>Usage</td>
<td>99</td>
</tr>
<tr>
<td>SolveigMM ASF Trimmer filter</td>
<td>102</td>
</tr>
<tr>
<td>Features</td>
<td>102</td>
</tr>
<tr>
<td>Usage</td>
<td>103</td>
</tr>
<tr>
<td>SolveigMM MPA Demuxer filter</td>
<td>95</td>
</tr>
<tr>
<td>Features</td>
<td>95</td>
</tr>
<tr>
<td>Usage</td>
<td>96</td>
</tr>
<tr>
<td>SolveigMM Trimmer filter</td>
<td>104</td>
</tr>
<tr>
<td>Features</td>
<td>105</td>
</tr>
<tr>
<td>Usage</td>
<td>105</td>
</tr>
<tr>
<td>SolveigMM VP8 Trimmer filter</td>
<td>106</td>
</tr>
<tr>
<td>Requirements</td>
<td>106</td>
</tr>
<tr>
<td>Features</td>
<td>106</td>
</tr>
<tr>
<td>Usage</td>
<td>107</td>
</tr>
<tr>
<td>SolveigMM AVC Trimmer filter</td>
<td>106</td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Features</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SolveigMM Vorbis Decoder filter</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM VP8 Decoder filter</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM VP8 Encoder filter</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM Silence Generator Filter</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM Frame Stepper</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM Audio Peak Reader</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM Thumbnails Generation Library</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SolveigMM Timeline ActiveX Control</td>
<td>Features</td>
<td>Usage</td>
</tr>
<tr>
<td>SDK Batch file structure</td>
<td>Batch samples</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch trimming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch joining</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch indexation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch extract video</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch extract audio</td>
<td></td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK Notices

SDK Version: 3.0 FULL
First edition: February 27, 2007
Date modified: September 12, 2013

More information can be found at: http://www.solveigmm.com/

For Technical Support, contact the Solveig Multimedia Technical Support Team: support@solveigmm.com

For purchasing and licensing information, contact the Sales Department: info@solveigmm.com
For more information, contact Solveig Multimedia Tel: +7 382 2 701 455 ext. 2049; Fax: +1 801 991-5443

Solveig Multimedia 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. Solveig Multimedia 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.

Solveig Multimedia,
Razvitiya ave. 3, 634055,
Tomsk, Russian Federation
www.solveigmm.com

© 2006 – 2013 Solveig Multimedia. All rights reserved.
Release Notes

Product Description

The SolveigMM Video Editing SDK is a software development kit to enable programmers to develop digital video/audio editing applications using Solveig Multimedia components. SolveigMM Video Editing Engine is a COM object to allow fast and lossless nonlinear video and audio editing. The trimming and joining technologies does not involve any decoding-encoding process.

SolveigMM Video Editing Engine supports the features as follows:

- **Multipart GOP-accurate trimming**
  - AVI file format (Audio Video Interleave) with video codecs: MP42, MP43, DX50, XVID, DIV3, DIVX, H263, MJPG, AVI DV type 1,2;
  - ASF format (Advanced Systems Format);
  - MPEG-2 format (Moving Pictures Expert Group);
  - MPEG-1 format (Moving Pictures Expert Group);
  - MP4 format (MPEG-4 Part 14);
  - MKV format (Matroska);
  - AVCHD format (Advanced Video Codec High Definition);
  - MOV format (QuickTime multimedia file format);
  - WMV format (Windows Media Video);
  - FLV format (Flash Video);
  - WEBM (HTML5 video) format.

- **Multipart frame accurate trimming**
  - AVI file format (Audio Video Interleave) with video codecs: MP42, MP43, DX50, XVID, DIV3, DIVX, H263, MJPG, AVI DV type 1,2;
  - MPEG-1 format (Moving Pictures Expert Group);
  - MP4 format (MPEG-4 Part 14);
  - MKV format (Matroska);
  - AVCHD format (Advanced Video Codec High Definition);
  - MOV format (QuickTime multimedia file format);
  - WMV format (Windows Media Video);
  - FLV format (Flash Video);
  - WEBM (HTML5 video) format.

- **Multipart trimming audio files**
  - MPEG-1,2 audio (Moving Pictures Expert Group audio format);
  - WMA format (Windows Media Audio);
  - WAV format (the Wave file format).

- **Re-multiplexing various media formats without re-encoding to**
  - ASF format (Advanced Systems Format);
  - MP4 format (MPEG-4 Part 14);
  - MOV format (QuickTime multimedia file format);
  - MKV format (Matroska);
  - FLV format (Flash Video);
  - WEBM (HTML5 video) format.

- **Media files joining**
  - AVI file format (Audio Video Interleave) with video codecs: MP42, MP43, DX50, XVID, DIV3, DIVX, H263, MJPG, AVI DV type 1,2;
  - ASF format (Advanced Systems Format);
- MPEG-2 format (Moving Pictures Expert Group);
- MPEG-1 format (Moving Pictures Expert Group);
- MP4 format (MPEG-4 Part 14);
- MKV format (Matroska);
- AVCHD format (Advanced Video Codec High Definition);
- MOV format (QuickTime multimedia file format);
- WMV format (Windows Media Video);
- MPEG-1,2 audio (Moving Pictures Expert Group audio format);
- WMA format (Windows Media Audio);
- WAV format (the Wave file format);
- WEBM (HTML5 video) format;
- FLV file format.

Currently supported mode - file by file joining. No reencoding involved, file should have equal parameters. For more information see SolveigMM Media Joiner Filter and Components description.

- Repairing, Indexing damaged or unindexed Windows Media Files (WMV, WMA, ASF files) by means of re-multiplexing
- Extracting audio or video streams from media files
- Processing batch of media files. All media files trimming configurations must be described within a batch file (*.xtl)
New features in SolveigMM Video Editing SDK 3.0

AVI files frame accurate trimming

The current feature is used to implement frame accurate trimming of AVI files with video codecs: MP42, MP43, DX50, XVID, DIV3, DIVX, H263, MJPG, AVI DV type 1,2.

The component that provides the feature is SolveigMM Trimmer Filter.

For more information see Components description and SMM Trim C++: AVI, ASF Sample.

Timeline ActiveX Control with enabled thumbnails and audio waveform

This component allows to use timeline with thumbnails and audio waveform for sample applications.

The component that provides the feature is SolveigMM TimeLine ActiveX control.

For more information see Components description.

Supporting FLV files frame/K-frame accurate trimming, joining and muxing

The SDK supports FLV file format frame accurate trimming with AVC1 video, other codecs trimmed with K-frame accuracy, joining and muxing.

The components that provide the feature are SolveigMM FLV Muxer Filter, SolveigMM FLV Splitter Filter, appropriate trimmer filter (depends on video codec), SolveigMM Media Joiner Filter.

For more details see SMM Trim C++, SolveigMM Media Joiner Filter, Join C++ Sample, SMM Mux C++: FLV Sample and Components description.

Supporting WEBM files frame/K-frame accurate trimming, joining and muxing

The SDK supports WEBM file format trimming, joining and muxing.

The components that provide the feature are SolveigMM VP8 Trimmer, SolveigMM VP8 Encoder, SolveigMM VP8 Decoder, SolveigMM Matroska Muxer, SolveigMM Matroska Filter, SolveigMM Media Joiner Filter.

For more details see SMM Trim C++, SolveigMM Media Joiner Filter, Join C++ Sample, SMM Mux C++: MKV, MP4, MOV, WEBM Sample and Components description.

SolveigMM Video Splitter sample

Video Editing SDK 3.0 includes now the sample of SolveigMM Video Splitter, which demonstrates of trimming and joining of all supported formats (see Product description).
Video Splitter displays the route on Yandex, Google maps and speedometer, compass, accelerometer readings for files of car video recorders that contain GPS extra data.

The sample is supplied with Timeline ActiveX Control with enabled thumbnails and audio waveform.

For more information please see SolveigMM Video Splitter.

**MP4 to ASF muxing for Microsoft Playready platform (supported by Galaxy S2)**

This sample demonstrates using SolveigMM ASF Muxer Filter within DirectShow Graph. It shows how to remux MP4 file to Microsoft PlayReady ASF file from command-line.

**Save Frame to JPEG, BMP sample**

This sample demonstrates the possibility to save any frame or group of frames to JPEG or BMP file.

The component that provide the feature is SolveigMM Thumbnails Generation Library.

For more information see SolveigMM Screenshot Grabber and Components description.
Components

SolveigMM Video SDK involves the DirectShow ® filters and COM components as follows

Top level COM objects:

- **SolveigMM Video Editing Engine (SMM_EditEngine.dll)**
  COM object allows frame/GOP accurate trimming and joining of MPEG-2, MPEG-1, AVI, MP4, MKV, MOV, AVCHD, FLV, WMV/ASF/WMA, MPEG audio files without decoding/encoding.

- **SolveigMM TimeLine ActiveX control (SMM_OCXSlider.ocx)**
  ActiveX time line control.

- **SolveigMM Custom Encoder (SMM_CustomEncoder.dll)**
  Library for partial reencoding of video fragments.

- **SolveigMM Frame Stepper (SMM_FrameStepper.dll)**
  COM object for frame/K-frame navigation on media files.

- **SolveigMM Audio Silence Generation Library (SMM_SilenceGen.ax)**
  COM object for gettings samples for muted audio with different compression format.

- **SolveigMM Audio Peak Reader (SMM_AudioPeakReader.dll)**
  COM object for getting amplitude peaks data for audio streams to display waveform.

- **SolveigMM Thumbnails Generation Library (SMM_ThumbNail.dll)**
  Library for generation thumbnails from video files.

DirectShow Trimmer filters:

- **SolveigMM Trimmer (SMM_Trimmer.ax)**
  DirectShow filter for GOP accurate AVI-compatible video streams trimming and MPEG1, 2 audio (ISO/IEC 11172-3, ISO/IEC 13818-3) streams trimming.

- **SolveigMM-Elecard MPEG2 Frame Accurate Trimmer (SMM_MP2TrimmerFA.ax)**
  DirectShow filter for MPEG-1/MPEG-2 video streams frame accurate trimming and compatible audio streams trimming.

- **SolveigMM AVCHD Frame Accurate Trimmer (SMM_MPEG4TrimmerFA.ax)**
  DirectShow filter for AVC/H264 video streams frame accurate trimming and compatible audio streams trimming, like MPEG1, MPEG2 audio, AC3 etc.

- **SolveigMM ASF Trimmer (SMMASF_Trimmer.ax)**
  DirectShow filter for frame and GOP accurate trimming of Windows Media Video and Audio streams.

- **SolveigMM VP8 Trimmer (SMM_VP8TrimmerFA.ax)**
  DirectShow filter for VP8 video streams GOP/frame accurate trimming.
DirectShow media format Demultiplexer/Multiplexer filters:

- **SolveigMM File Writer (SMM_FileWriter.ax)**
  DirectShow filter for writing AVI and MPEG audio files. It is intended for keeping A/V synchronization during AVI trimming with MPEG VBR audio.

- **SolveigMM ASF Multiplexer (SMM_ASFMuxer.ax)**
  DirectShow filter for the video and audio streams multiplexing into Microsoft Advanced Streaming Format. It is based on Microsoft Windows Media Format SDK and allows Windows Media video/audio writing and video/audio streams encoded by third-party codec like MPEG-4 AVC, DivX, XviD,MPEG-2 Video, AC3, MPEG1,2 audio etc.

- **SolveigMM Matroska Muxer (SMM_MKVMuxer.ax)**
  DirectShow filter for the video, audio and subtitles streams multiplexing into Matroska Format. It is based on libmatroska and libebml LGPL libraries.

- **SolveigMM Matroska Splitter (SMM_MKVSplitter.ax)**
  DirectShow filter for demultiplexing audio, video and subtitles from Matroska Format. It is based on libmatroska and libebml LGPL libraries.

- **SolveigMM FLV format Demultiplexer (SMM_FLVDemuxer.ax)**
  DirectShow filter for demultiplexing FLV format.

- **SolveigMM FLV format Multiplexer (SMM_FLVMuxer.ax)**
  DirectShow filter for video and audio streams multiplexing into FLV file format.

- **SolveigMM MP4 format Demultiplexer (SMM_MP4Demuxer.ax)**
  DirectShow filter for demultiplexing QuickTime File Format and MPEG-4 Part 14 streams.

- **SolveigMM MP4 format Multiplexer (SMM_MP4Muxer.ax)**
  DirectShow filter for video and audio streams multiplexing into QuickTime File Format and MPEG-4.

- **SolveigMM MPA format Demultiplexer(SMM_MPADemuxer.ax)**
  DirectShow filter for demultiplexing audio from MPEG audio format, and accurate navigation on Variable Bitrate MPEG audio files.

DirectShow Decoder/Encoder filters:

- **SolveigMM VP8 Encoder (SMM_VP8Encoder.ax)**
  DirectShow filter for encoding VP8 video. It is based on WebM project source code.

- **SolveigMM VP8 Decoder (SMM_VP8Decoder.ax)**
  DirectShow filter for decoding VP8 video. It is based on WebM project source code.

- **SolveigMM Vorbis Decoder (SMM_VorbisDecoder.ax)**
  DirectShow filter for decoding Vorbis Audio. It is based on WebM project source code.

Other DirectShow filters:

- **SolveigMM Media Joiner (SMM_MediaJointer.ax)**
  DirectShow filter for joining video/audio streams of different files. Currently supported mode: whole file-by-file joining of MPEG audio, AVI, DV-AVI Type 1 and 2, WMA, WMV, ASF, MPEG-1.
The filter does not support reencoding, so the chosen files' critical parameters must be the same. For more information on how to validate parameters see `SMAT_ValidateFiles`.

- **SolveigMM K Frame Manager (SMM_KFrManager.ax)**
  DirectShow filter for providing K frame navigation during a playback.

- **SolveigMM BMP Push Source (SMM_BMPPushSrc.Ax)**
  DirectShow filter to make a video stream from chosen BMP files. Currently supported BMP format - 24/32 bit.

Other Dynamic Link Libraries:

- **SolveigMM ASF Validator (SMM_ASFValidator.dll)**
  Library for Windows Media Video files validation.

- **SolveigMM BMP To AVI (BmpToAvi.dll)**
  Based on SolveigMM BMP Push Source, DLL provides the routines to create AVI-file from the BMP pictures.
## Sample applications

### C++ Samples

#### C++ Trimming Samples:

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trimm_AVI_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features GOP accurate AVI multipart trimming</td>
</tr>
</tbody>
</table>
| SMM_Trim_AVCHD_C++      | This sample application demonstrates usage of SolveigMM MPEG-4 Frame Accurate Trimmer DirectShow Filter. It illustrates frame accurate trimming of AVC/H264 streams wrapped in the Program or Transport streams. It requires the following DirectShow filters:  
  - Elecard AVC Video Encoder  
  - Elecard AVC Video Decoder  
  - Elecard MPEG Demultiplexer  
  - Elecard Multiplexer  
  - Elecard Index Reader         |
| SMM_TrimMOV_C++         | This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate MOV multipart trimming          |
| SMM_Trim_MP4_C++        | This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate MP4 multipart trimming          |
| SMM_Trim_MKV_C++        | This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate MKV multipart trimming          |
| SMM_Trim_MPEG1_C++      | This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MPEG 1,2 audio trimming                      |
| SMM_Trim_MPEG2_C++      | This sample application demonstrates using SolveigMM-Elecard MPEG-2 Frame Accurate Trimmer DirectShow Filter. It illustrates frame accurate multipart trimming MPEG-2 files. It requires the following DirectShow filters:  
  - Elecard MPEG-2 Video Encoder  
  - Elecard MPEG-2 Video Decoder  
  - Elecard MPEG Demultiplexer  
  - Elecard Multiplexer  
  - Elecard Index Reader |
## SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trim_FLV_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate FLV multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_WEBM_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate WEBM multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_MP3_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MP3 audio trimming</td>
</tr>
<tr>
<td>SMM_Trim_WAV_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features Wave audio trimming</td>
</tr>
<tr>
<td>SMM_Trim_WMA_C++</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features Windows Media audio trimming</td>
</tr>
</tbody>
</table>

### C++ Multiplexing Samples

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Mux_ASF_C++</td>
<td>This sample demonstrates using SolveigMM ASF Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio streams encoded by third party codecs into Microsoft ASF format. It supports AVI to ASF re-multiplexing, WMV to ASF re-multiplexing, WMA 2 ASF re-multiplexing, MPEG-2 Program or Transport stream to ASF re-multiplexing, MPEG-4 AVC video contained in MP4 or MPEG-2 TS format to ASF re-multiplexing, MPEG-1,2 audio to ASF re-multiplexing.</td>
</tr>
<tr>
<td>SMM_Mux_MKV_C++</td>
<td>This sample demonstrates using SolveigMM Matroska Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into Matroska format.</td>
</tr>
<tr>
<td>SMM_Mux_MOV_C++</td>
<td>This sample demonstrates using SolveigMM MP4 format Multiplexer within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into QuickTime multimedia file format.</td>
</tr>
<tr>
<td>SMM_Mux_MP4_C++</td>
<td>This sample demonstrates using SolveigMM MP4 format Multiplexer within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into MPEG-4 Part 14 format.</td>
</tr>
<tr>
<td>SMM_Mux_FLV_C++</td>
<td>This sample demonstrates using SolveigMM FLV format Multiplexer within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into FLV format.</td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK

SMM_Mux_WEBM_C++
This sample demonstrates using SolveigMM WEBM format Multiplexer within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into WEBM format. The supported video codec is VP8, audio codec - Vorbis.

C++ Joining Samples

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Join_AVI_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of AVI files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_JoinASF_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of ASF files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_AVCHD_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of AVCHD files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MKV_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MKV files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MOV_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MOV files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MP3_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MP3 files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MP4_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MP4 files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>Sample Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SMM_Join_FLV_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of FLV files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MPEG1_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MPEG1 files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MPEG2_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of MPEG2 files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_WAV_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of WAV files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_WMA_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of WMA files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_WEBM_C++</td>
<td>The sample application is to show the most significant steps to perform joining operation of WEBM files format. The sample is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
</tbody>
</table>

**Other C++ Samples**

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_ASFMarkers</td>
<td>This sample application demonstrates using SolveigMM Video Editing Engine. It illustrates ASF markers workflow.</td>
</tr>
<tr>
<td>SMM_DShowAVITrim</td>
<td>This sample demonstrates using SolveigMM Trimmer Filter within DirectShow Graph. It shows how to trim AVI files.</td>
</tr>
<tr>
<td>SMM_BatchSplit</td>
<td>This console application demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features - supporting batch of</td>
</tr>
</tbody>
</table>
media files trimming. All media files trimming configurations must be described within a batch file (*.xtl) to be an input parameter of BatchSplit sample application.

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMM_MP4ToASF</strong></td>
<td>This sample demonstrates using SolveigMM AVI Muxer Filter within DirectShow Graph. It shows how to trim the AVI file from command-line.</td>
</tr>
<tr>
<td><strong>SMM_VSplitter</strong></td>
<td>This application demonstrates using SolveigMM Video Editing Engine and ActiveX time line control. It also demonstrates displaying route on Yandex, Google maps and speedometer, compass, accelerometer readings, displaying extra camera view on avto DVR media files.</td>
</tr>
<tr>
<td><strong>SMM_Screensot_Grabber_C++</strong></td>
<td>This sample application demonstrates using SolveigMM Thumbnails Generation Library. It allows to get the screenshot or the sequence of screenshots on the selected time interval.</td>
</tr>
</tbody>
</table>

### C# Samples

**C# Trimming Samples:**

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMM_Trim_AVI_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features AVI files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_AVCHD_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features AVCHD files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_TrimASF_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features ASF files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_MKV_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MKV files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_MOV_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MOV files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_MP3_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MP3 files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_MP4_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MP4 files multipart trimming</td>
</tr>
<tr>
<td><strong>SMM_Trim_FLV_C#</strong></td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features FLV files multipart trimming</td>
</tr>
<tr>
<td>Name of Sample</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SMM_Trim_MPEG1_C#</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MPEG1 files multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_MPEG2_C#</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MPEG2 files multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_WAV_C#</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features WAV files multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_WMA_C#</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features WMA files multipart trimming</td>
</tr>
<tr>
<td>SMM_Trim_WEBM_C#</td>
<td>This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features WEBM files multipart trimming</td>
</tr>
</tbody>
</table>

**C# Joining Samples**

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Join_ASF_C#</td>
<td>The sample application is to show the most significant steps to perform joining operation of ASF. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_AVCHD_C#</td>
<td>The sample application is to show the most significant steps to perform joining operation of AVCHD files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_AVI_C#</td>
<td>The sample application is to show the most significant steps to perform joining operation of AVI files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MKV_C#</td>
<td>The sample application is to show the most significant steps to perform joining operation of MKV files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
<tr>
<td>SMM_Join_MOV_C#</td>
<td>The sample application is to show the most significant steps to perform joining operation of MOV files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).</td>
</tr>
</tbody>
</table>
The sample application is to show the most significant steps to perform joining operation of MP3 files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of MP4 files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of FLV files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of MPEG2 files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of WAV files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of WMA files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

The sample application is to show the most significant steps to perform joining operation of WEBM files. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).

### Other C# Samples

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_BMPToAVIUtility</td>
<td>The .NET C# sample application performs using BMPToAVI.dll to</td>
</tr>
</tbody>
</table>
produce the AVI video file from BMPs.

## Delphi Samples

### Delphi Trimming Samples:

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trim_ASF_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of ASF files.</td>
</tr>
<tr>
<td>SMM_Trim_AVCHD_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of AVCHD files.</td>
</tr>
<tr>
<td>SMM_Trim_AVI_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of AVI files.</td>
</tr>
<tr>
<td>SMM_Trim_MKV_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MKV files.</td>
</tr>
<tr>
<td>SMM_Trim_MOV_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MOV files.</td>
</tr>
<tr>
<td>SMM_Trim_MP3_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MP3 files.</td>
</tr>
<tr>
<td>SMM_Trim_MP4_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MP4 files.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG1_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MPEG1 files.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG2_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MPEG2 files.</td>
</tr>
<tr>
<td>SMM_Trim_WAV_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WAV files.</td>
</tr>
<tr>
<td>SMM_Trim_WMA_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WMA files.</td>
</tr>
<tr>
<td>SMM_Trim_WEBM_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WEBM files.</td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trim_FLV_Delphi</td>
<td>This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of FLV files.</td>
</tr>
</tbody>
</table>

**Delphi Joining Samples:**

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Join_ASF_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining ASF files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_AVCHD_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining AVCHD files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_AVI_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining AVI files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MKV_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MKV files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MOV_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MOV files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MP3_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MP3 files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MP4_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MP4 files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MPEG1_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MPEG1 files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_MPEG2_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining MPEG2 files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_WAV_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining WAV files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_WMA_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining WMA files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_WEBM_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining WEBM files using Delphi language</td>
</tr>
<tr>
<td>SMM_Join_FLV_Delphi</td>
<td>This is a Delphi analogue of C++ Joiner sample. It demonstrates how to perform joining FLV files using Delphi language</td>
</tr>
</tbody>
</table>
## VB.60 Samples

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trim_ASF_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of ASF files.</td>
</tr>
<tr>
<td>SMM_Trim_AVCHD_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of AVCHD files.</td>
</tr>
<tr>
<td>SMM_Trim_AVI_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of AVI files.</td>
</tr>
<tr>
<td>SMM_Trim_MKV_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MKV files.</td>
</tr>
<tr>
<td>SMM_Trim_MOV_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MOV files.</td>
</tr>
<tr>
<td>SMM_Trim_MP3_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MP3 files.</td>
</tr>
<tr>
<td>SMM_Trim_MP4_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MP4 files.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG1_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MPEG1 files.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG2_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of MPEG2 files.</td>
</tr>
<tr>
<td>SMM_Trim_WAV_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WAV files.</td>
</tr>
<tr>
<td>SMM_Trim_WMA_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WMA files.</td>
</tr>
<tr>
<td>SMM_Trim_WEBM_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of WEBM files.</td>
</tr>
<tr>
<td>SMM_Trim_FLV_VB60</td>
<td>The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of FLV files.</td>
</tr>
</tbody>
</table>
### VB.Net Samples

#### VB.Net Trimming Samples:

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_Trim_ASF_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_AVCHD_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_AVI_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MKV_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MOV_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MP3_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MP4_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG1_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_MPEG2_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_WAV_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_WMA_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_WEBM_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
<tr>
<td>SMM_Trim_FLV_VBNet</td>
<td>It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.</td>
</tr>
</tbody>
</table>
**VB.Net Joining Samples:**

<table>
<thead>
<tr>
<th>Name of Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM_JoinASF_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining ASF files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinAVCHD_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining AVCHD files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinAVI_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining AVI files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMKV_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MKV files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMOV_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MOV files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMP3_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MP3 files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMP4_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MP4 files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMPEG1_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MPEG1 files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinMPEG2_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining MPEG2 files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinWAV_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining WAV files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinWMA_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining WMA files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinWEBM_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining WEBM files using VB.Net language</td>
</tr>
<tr>
<td>SMM_JoinFLV_VBNet</td>
<td>This is a VB.Net analogue of SimpleJoiner sample. It demonstrates how to perform joining FLV files using VB.Net language</td>
</tr>
</tbody>
</table>
System requirements

- CPU (Intel® Pentium II, Celeron, AMD® Athlon, Opteron etc.)
- 128 MB RAM.
- Any VGA card.
- Windows® XP / 2000 / 2003 / Vista / 7 / 8
- Microsoft® DirectX® Media SDK or Microsoft® DirectX® 8.1 SDK.
- Appropriate video and audio decoders for preview feature
- Microsoft® Windows Media Format 9 Series Runtime® (to be shipped with Windows Media Player 9 ® and higher)
- Windows Media Format 11 Series Runtime for WMV file frame accurate editing
User Guide

SDK folder structure

After installing the SDK, the “SolveigMM Video Editing SDK” folder will appear in the destination folder specified during installation. 

**Note:** After installing the SolveigMM Video Editing SDK, the SDK Components (filters and DLLs) will be installed on your PC in the “Program Files\Common Files\Solveig Multimedia” (in addition to SolveigMM Video Editing SDK folder).

The SDK folder has the following folders:

- **Bin** - contain release versions of compiled samples
- **deps** - contain Elecard SDK components related files
  - **deps\elecard_codecs_sdk** - includes Elecard components and settings IDs
  - **deps\elecard_sdk** - includes Elecard components and settings IDs
  - **deps\elecard_sdk\Classes** - includes Elecards SDK headers to be used by the samples for creating DirectShow filter graph only
  - **deps\elecard_sdk\lib** - includes Elecard SDK debug\release libraries to be used by the samples for creating DirectShow filter graph only (in Visual Studio versions 7 and 7.1). This is a default library directory.
  - **deps\elecard_sdk\libVS2005** - includes Elecard SDK debug\release libraries to be used by the samples for creating DirectShow filter graph only (in Visual Studio versions 8).
  - **deps\elecard_sdk\libVS2008** - includes Elecard SDK debug\release libraries to be used by the samples for creating DirectShow filter graph only (in Visual Studio versions 9 and higher).

- **Doc** - includes all SDK-related documentation
- **Include** - includes headers to comprise interfaces and GUIDS headers
- **Samples** – includes VC++ 7.1, VC# VB.NET, VB 6.0 Sample Application and media files

Installing SDK

1. To install the SolveigMM Video Editing SDK:
2. Run the SDK setup. To run, double click the executable file from the SDK setup package.
3. The SolveigMM Video Editing SDK window will appear. Read the recommendations and warnings. Click Next.
5. 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.
6. Select the destination folder in which you want to install the SDK. Click Next.
7. Select the program group in which you want the SDK to be located. Click Next.
8. To complete an 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 “SolveigMM Video Editing SDK has been successfully installed” will appear and the SDK is ready to use.

Uninstalling SDK

To uninstall the SolveigMM Video Editing SDK:

1. Click -> Start -> Programs -> Solveig Multimedia -> SolveigMM Video Editing SDK xx -> Uninstall SDK
   (xx – the SDK version number).

Follow the onscreen instructions to complete removal of the application. **Note** that you might need to remove some files manually after the uninstallation.
SDK Sample Applications

C++ Samples

C++ Trimming Samples:

SMM Trim C++: AVI, ASF

This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features frame accurate ASF and AVI multiparts trimming.

SMM Trim C++: MP3, WMA, WAV

This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MP3, WMA, WAV trimming.
SMM Trim C++: MPEG1, MPEG2

This sample application demonstrates using SolveigMM-Elecad MPEG-2 Frame Accurate Trimmer DirectShow Filter. It illustrates frame accurate multipart trimming MPEG-2, MPEG-1 files. It requires the following DirectShow filters:

- Elecard MPEG-2 Video Encoder
- Elecard MPEG-2 Video Decoder
- Elecard MPEG Demultiplexer
- Elecard Multiplexer
- Elecard Index Reader

SMM Trim C++: AVCHD, MP4, MOV, FLV, MKV, WEBM

This sample application demonstrates usage of SolveigMM AVC Trimmer FA DirectShow Filter. It illustrates frame accurate trimming of AVCHD, MP4, MOV, FLV, MKV, WEBM file formats. It requires the following DirectShow filters:

- Elecard AVC Video Encoder
- Elecard AVC Video Decoder
- Elecard MPEG Demultiplexer
- Elecard Multiplexer
- Elecard Index Reader
**C++ Multiplexing Samples**

**SMM Mux C++: ASF**

This sample demonstrates using SolveigMM ASF Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio streams encoded by third party codecs into Microsoft ASF format. It supports AVI to ASF re-multiplexing, WMV to ASF re-multiplexing, WMA to ASF re-multiplexing, MPEG-2 Program or Transport stream to ASF re-multiplexing, MPEG-4 AVC video contained in MP4 or MPEG-2 TS format to ASF re-multiplexing, MPEG-1,2 audio to ASF re-multiplexing.
SolveigMM Video Editing SDK

Solveig Multimedia

This sample demonstrates using SolveigMM Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into Matroska, MP4, MOV and WEBM format.

For WEBM format the supported video codec is VP8, audio codec is Vorbis.

**SMM Mux C++: MKV, MP4, MOV, WEBM**

This sample demonstrates using SolveigMM Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio/subtitles streams encoded by third party codecs into Matroska, MP4, MOV and WEBM format.

For WEBM format the supported video codec is VP8, audio codec is Vorbis.
SMM Mux C++: FLV

This sample demonstrates using SolveigMM FLV Muxer Filter within DirectShow Graph. It shows how to multiplex video/audio streams encoded by third party codecs into FLV format.
**C++ Joining Samples**

**SMM Join C++: ASF, AVCHD, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, FLV, WAV, WMA, WEBM**

The sample application is to show the most significant steps to perform joining operation of ASF, AVCHD, AVI, MKV, MOV, FLV, MP3, MP4, MPEG1, MPEG2, FLV, WAV, WMA, WEBM file formats.

![SMM Join C++](image)

**Other C++ Samples**

**SolveigMM Batch Splitter Sample Application**

This console application demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features - supporting batch of media files trimming. All media files trimming configurations must be described within a batch file (*.xtl) to be an input parameter of BatchSplit sample application.

**SolveigMM ASF Markers**

This console application demonstrates using SolveigMM Video Editing Engine. It illustrates ASF markers workflow – add a marker to ASF, remove particular marker from ASF, list all the markers from input ASF.

**SolveigMM AVI Trimmer Console Utility**

This sample demonstrates using SolveigMM AVI Muxer Filter within DirectShow Graph. It shows how to trim the AVI file from command-line.

**SolveigMM MP4 to ASF for PlayReady**

This sample demonstrates using SolveigMM ASF Muxer Filter within DirectShow Graph. It shows how to remux MP4 file to Microsoft PlayReady ASF file from command-line.
SolveigMM Video Splitter

This sample application demonstrates using SolveigMM Video Editing Engine and ActiveX time line control. It also demonstrates displaying route on Yandex, Google maps and speedometer, compass, accelerometer readings, displaying extra camera view on avto DVR media files.

SolveigMM Screenshot Grabber

This sample application demonstrates using SolveigMM Thumbnails Generation Library. It allows to get the screenshot or the sequence of screenshots on the selected time interval.
C# Samples

C# Trimming Samples:

SMM Trim C#: ASF, AVCHD, AVI, MKV, MOV, FLV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM

This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features ASF, AVCHD, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, FLV, WAV, WMA and WEBM files multipart trimming.
C# Joining Samples

SMM Join C#: ASF, AVCHD, AVI, MKV, MOV, FLV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM

The sample application is to show the most significant steps to perform joining operation. Joiner is based on SolveigMM Video Editing Engine, which uses SolveigMM Media Joiner filter and meets its restrictions (see SMM MediaJoiner description).
Other C# Samples

SolveigMM BMPs to AVI Utility

The .NET C# sample application performs using BMPToAVI.dll to produce the AVI video file from BMPs.

Delphi Samples

Delphi Trimming Samples:

SMM Trim Delphi: ASF, AVCHD, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM, FLV

This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of files that have supported formats
SolveigMM Video Editing SDK

**Delphi Joining Samples:**

**SMM Trim Delphi**: ASF, AVCHD, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM, FLV

This is a Delphi analogue of SimpleJoiner sample. It demonstrates how to perform joining using Delphi language.

![SMM Join Delphi](image)

**VB.60 Samples**

**SMM Trim VB6.0**: ASF, AVCHD, FLV, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM

The VB6.0 sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of files with supported formats.
vb.net samples

vb.net trimming samples

SMM Trim VB.Net: ASF, AVCHD, FLV, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM

It is similar to VB 6.0 trim sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.
SolveigMM Video Editing SDK

VB.Net Joining Samples

SMM Join VB.Net: ASF, AVCHD, FLV, AVI, MKV, MOV, MP3, MP4, MPEG1, MPEG2, WAV, WMA, WEBM

It is similar to VB 6.0 joiner sample but is intended to show the usage of EditingEngine in .Net environment with VisualBasic language.
SolveigMM Video Editing SDK Reference Manual

SolveigMM Video Editing Engine

SolveigMM Video Editing Engine COM is the basic SDK’s object. It’s initiates appropriate graphs to perform the requested operations on chosen files. It also manages the other components included in SDK and provides the methods and interfaces to control the chosen task. Video Editing Engine allows frame/GOP accurate trimming and joining of MPEG1, MPEG-2, AVI, AVCHD, MKV, MP4, MOV, FLV, WEBM, WMV/WMA/ASF, WAV, MPEG audio files.

Editing Engine enumerations

SMM_OutputTypes

Indicates the type of the file.

```c
enum SMM_OutputTypes {
    SMM_File_Type_NO = 0x0,
    SMM_File_Type_AVI = 0x1,
    SMM_File_Type_ASF = 0x2,
    SMM_File_Type_MPA = 0x3,
    SMM_File_Type_MPEG2_PS = 0x4,
    SMM_File_Type_MPEG2_TS = 0x5,
    SMM_File_Type_MPEG2_VES = 0x6,
    SMM_File_Type_MPEG1_SYS = 0x7,
};
```

Elements:

SMM_File_Type_NO
Type not specified.

SMM_File_Type_AVI
Audio video interleaved format.

SMM_File_Type_ASF
Advanced streaming format (includes *.asf, *.wmv, *.wma files).

SMM_File_Type_MPA
MPEG audio (MPEGv1,2 layers1-3).

SMM_File_Type_MPEG2_PS
MPEG-2 Program Stream.

SMM_File_Type_MPEG2_TS
MPEG-2 Transport Stream.

SMM_File_Type_MPEG2_VES
MPEG-2 Video Elementary Stream.

SMM_File_Type_MPEG1_SYS
MPEG-1 System stream.
SMM_TaskType

The type of process to perform on chosen files.

```
enum SMM_TaskType {
    SMM_Task_Type_Trimming = 0x0,
    SMM_Task_Type_Joining = 0x1,
    SMM_Task_Type_Join_Gap_Fill = 0x2,
    SMM_Task_Type_Muxing = 0x3,
    SMM_Task_Type_Demuxing = 0x4,
}
```

**Elements:**

**SMM_Task_Type_Trimming**
trimming process is planned.

**SMM_Task_Type_Joining**
joining process is planned.

**SMM_Task_Type_Join_Gap_Fill**
currently not used.

**SMM_Task_Type_Muxing**
currently not used.

**SMM_Task_Type_Demuxing**
currently not used.

SMM_TrimmAccuracyType

The desired accuracy of trimming

```
enum SMM_TrimmAccuracyType {
    SMM_TrimmAccuracyType_Deflt = -1,
    SMM_TrimmAccuracyType_GOP = 0,
    SMM_TrimmAccuracyType_Frame = 1
};
```

**Elements:**

**SMM_TrimmAccuracyType_Deflt**
type not set.

**SMM_TrimmAccuracyType_GOP**
GOP-accurate trimming.

**SMM_TrimmAccuracyType_Frame**
Frame-accurate trimming.

SMM_TrimFlags

Trimming settings and parameters

```
enum SMM_TrimFlags {
    SMM_TrimFlags_None = 0,
```

…
SolveigMM Video Editing SDK

```c
SMM_TrimFlags_ObeySampleTimes = 0x2,
SMM_TrimFlags_VideoOnly = 0x8,
SMM_TrimFlags_AudioOnly = 0x10,
SMM_TrimFlags_GettingSize = 0x20,
SMM_TrimFlags_InASF = 0x40,
SMM_TrimFlags_UsePositioning = 0x80,
SMM_TrimFlags_IgnoreTrimErrors = 0x100,
SSM_TrimFlags_TrimOverlappedIntervals = 0x200
};
```

**Elements:**

**SMM_TrimFlags_None**

**SMM_TrimFlags_ObeySampleTimes**
Use alternative method of audio/video synchronization (by default is not set).

**SMM_TrimFlags_VideoOnly**
Process a video stream only regardless of a number video and audio stream an input media file contains. The engine processes a first video stream if finds. As an alternative you can use SMAT_VideoStreamNumber parameter.

**SMM_TrimFlags_AudioOnly**
Process an audio stream only regardless of a number video and audio stream an input media file contains. The engine processes a first audio stream if finds. As an alternative you can use SMAT_AudioStreamNumber parameter.

**SMM_TrimFlags_GettingSize**
Indicates whether getting file size not trimming file. Does not write an output file to HDD.

**SMM_TrimFlags_InASF**
Used by the video editing engine.

**SMM_TrimFlags_UsePositioning**
Use an alternative trimming method. The method faster then default one. For the present it is recommended to set this flag in one-part trimming mode of AVI, MP3, WMA files.

**SMM_TrimFlags_IgnoreTrimErrors**
Do not obey system errors during trimming process.

**SSM_TrimFlags_TrimOverlappedIntervals**
After trimming of each interval trim process stops and when starts from the beginning of the input media file.

**Editing Engine structures**

**SMM_GROUP_TASK( Structure )**

**Description:**
Describes the group part of the task (See group element of SDK Batch file structure).

```c
struct SMM_GROUP_TASK {
CString szDstName ;
SMM_TaskType nTaskType ;
SMM_TRACK_INFO tracks[10] ;
UINT nTrackCount;
};
```
Members:

szDstName
Output file name. No default value.

nTaskType
Task type to be performed to get current output file. Default value: SMM_Task_Type_Trimming.

tracks
Track information for current group (See below).

nTrackCount
Number of tracks in current group. Default value: 1.

SMM_TRACKINFO (Class)

Description:
Describes the track part of the task (See track element of SDK Batch file structure).

class SMM_TRACKINFO {
public:
  int nVideoStream;
  int nAudioStream;
  WCHAR szMPEG2IdxFileName[1024];
  SMM_CLIPINFO *pClips;
  int nClips;
  SMM_OutputTypes nOutType;
  SMM_TrimmAccuracyType nTrimmAccurType;
  DWORD ulFlags;
  SMM_AutoSplitType dwASMode;
  LONGLONG llASValue;
  UINT nTrackPos;
  DWORD dwReserv[20];

  void CreateClips( int nNumber );
  void ClearClips ( void );
  void CopyFrom ( SMM_TRACKINFO TrackFrom );
};

Members:

nVideoStream
Number of video streams in files. Default value: 0.

nAudioStream
Number of audio streams in files. Default value: 0.
**szMPEG2IdxFileName**
MPEG2 Index file name.

**pClips**
Pointer to the clips information of the current track. Default value: NULL.

**nClips**
Quantity of clips in current track. Default value: 0.

**nOutType**
The type of the output file. Default value: SMM_File_Type_NO.

**nTrimmAccurType**
Trim accuracy type used (applied for Trimming only, ignored for joining). Default value: SMM_TrimmAccuracyType_Deflt.

**ulFlags**
Trimming flags (applied for Trimming only, ignored for joining).

**dwASMode**
Not used.

**IIASValue**
Not used.

**Member functions:**

**CreateClips**
Sets the quantity of SMM_CLIP_INFO structures and allocates memory to store the given number of clips. The allocated memory should always be released.

**ClearClips**
Releases the memory allocated by CreateClips() function and resets the nClips member.

**CopyFrom**
Copies data from the existing track.

---

**SMM_CLIP_INFO (Structure)**

**Description:**
Describes the exact clip part of the current track (See clip part of SDK Batch file structure.)

```c
struct SMM_CLIP_INFO {
    DWORD dwClipNum;
    REFERENCE_TIME rtStart;
    REFERENCE_TIME rtEnd;
    BOOL bMute;
    WCHAR wsfName[1024];
    DWORD dwReserv[20];
};
```

**Members:**

**dwClipNum**
The number of the current clip.
rtStart
Start time of the current clip (relative to the file's time).

rtEnd
End time of the current clip (relative to the file's time).

wsfName
The name of file, assigned to the current clip.

bMute
Not applied.

Notes:
If you're going to use the SMM_CLIP_INFO to perform the joining operation, you should set rtStart and rtEnd to 0, that means you are going to append the whole file to the output. This is currently the only mode for joining, otherwise the error will occure!

TrimInfoList( Structure )

Description:
Used to set trimming parameters to SolveigMM Video Editing Engine through SMAT_TrimList parameter.

```c
struct TrimInfoList
{
    public:
        UINT nListSize;
        __TSTrimInfo* trim_list;

    TrimInfoList( void );
    TrimInfoList( UINT size );
    void Clear( void );
};
```

Members:

nListSize
A count of fragments to be kept in output media file. Fragments start/stop values have to be described in __TSTrimInfo structures.

trim_list
A pointer to __TSTrimInfo array size of nListSize * sizeof (__TSTrimInfo)

Member functions:

TrimInfoList( void )
Default constructor.

TrimInfoList( UINT size )
Constructor, allocates memory for "size" trim_lists.
Clear
Releases memory allocated.

__TSTrimInfo( Structure )

Description:
Describes start/stop time values of media fragments to be kept and saved to output file.

typedef struct __tagTSTrimInfo {
    INT16 nPieceNum;
    LONGLONG llStartPos;
    LONGLONG llStopPos;
} __TSTrimInfo;

Members:

nPiecenum
A number of a fragment.

llStartPos
Start fragment time value in 100 nanoseconds units.

llStopPos
Stop fragment time value in 100 nanoseconds units.

FILE_VERSIONS_2( Structure )

Description:
Used to get objects being used by the Video Editing Engine friendly names and versions through
SMAT_GetVersions2 parameter.

struct FILE_VERSIONS_2 {
    int count;
    VERSION_INFO_2* data;
};

Members:

count
Count of objects to be used by the Video Editing Engine;

data
A pointer to `VERSION_INFO_2` array size of `count * sizeof (VERSION_INFO_2)`

---

**VERSION_INFO_2( Structure )**

**Description:**
Gives the information on filters used.

```c
struct VERSION_INFO_2 {
    WCHAR filename[100];
    WCHAR fileversion[100];
    WCHAR filepath[1024];
    DWORD dwReserved[100];
};
```

**Members:**

- **filename**
  A friendly name of an object being used by the Video Editing Engine.

- **fileversion**
  The version of the object.

- **filepath**
  The path to the current object.

---

**SMM_ASF_MARKER( Structure )**

**Description:**
Used to set/get certain ASF marker through SMAT_ASFMarker parameter.

```c
struct SMM_ASF_MARKER {
    WCHAR name[5120];
    REFERENCE_TIME time_pos;
    ULONG num_pos;
};
```

**Parameters:**

- **name**
  A marker name to be limited by 5120 wide characters.

- **time_pos**
  A marker time position in 100 nanosecond units.

- **num_pos**
  A zero-based index of a markers contained in ASF.

---

**TrimListSilenceNums( Class )**

**Description:**
Contains the numbers of intervals to be replaced with silence.
class SMM_TRACK_INFO {
    int* silenceNums;
    int count;
    TrimListSilenceNums ();
    TrimListSilenceNums (const TrimListSilenceNums& their);
    ~TrimListSilenceNums ();
    void resize (UINT size);
    void CopyFrom (const TrimListSilenceNums* list);
    void Clear ();
};

Members:

silenceNums
Array of intervals numbers to be replaced with silence.

count
Number of elements in the silenceNum array.

Member functions:

TrimListSilenceNums
Constructors of the class, that either creates a new array or copies the array from existing.

TrimListSilenceNums
Destructor of the class, deletes the initialized array member.

resize
Reallocates the array of specified size.

CopyFrom
Copies data from the existing TrimListSilenceNums class.

clear
deletes the allocated array of intervals, if any.

Editing Engine Interfaces

The following section describes the interfaces that are used to tune and control the SolveigMM Video Editing Engine.

IModuleConfig interface

The IModuleConfig interface provides access for the reading and writing of the module (SolveigMM Video Editing Engine) parameters. All of the module parameters have their own unique GUIDs. The parameter GUID must be known in order to read or write its value. All of the parameter values are transferred via the VARIANT structure that represents a general store for different data types.

For more detailed information about IModuleConfig interface please read Elecard Module Configuration Programmer Guide to be shipped with SolveigMM Video Editing SDK.

The GUIDs of parameters supported by SolveigMM Video Editing Engine through IModuleConfig interface are described in Editing Engine Parameters section.

IID of IModuleConfig interface is
### IID _IModuleConfig
{486F726E-4D43-49b9-8A0C-C22A2B0524E8}

### Methods in VTable order

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IUnknown</strong></td>
<td></td>
</tr>
<tr>
<td>QueryInterface</td>
<td>Retrieves pointers to supported interfaces</td>
</tr>
<tr>
<td>AddRef</td>
<td>Increments reference count</td>
</tr>
<tr>
<td>Release</td>
<td>Decrements reference count</td>
</tr>
<tr>
<td><strong>IPersist</strong></td>
<td></td>
</tr>
<tr>
<td>GetClassID</td>
<td>Retrieves the class identifier (CLSID) of an object</td>
</tr>
<tr>
<td><strong>IPersistStream</strong></td>
<td></td>
</tr>
<tr>
<td>IsDirty</td>
<td>Checks the object for changes since it was last saved</td>
</tr>
<tr>
<td>Load</td>
<td>Initializes an object from the stream where it was previously saved</td>
</tr>
<tr>
<td>Save</td>
<td>Saves an object into the specified stream and indicates whether the object should reset its dirty flag.</td>
</tr>
<tr>
<td>GetSizeMax</td>
<td>Return the size in bytes of the stream needed to save the object</td>
</tr>
<tr>
<td><strong>IModuleConfig</strong></td>
<td></td>
</tr>
<tr>
<td>SetValue</td>
<td>Sets a new parameter value.</td>
</tr>
<tr>
<td>GetValue</td>
<td>Retrieves the current parameter value.</td>
</tr>
<tr>
<td>GetParamConfig</td>
<td>Retrieves the pointer to the IParamConfig interface.</td>
</tr>
<tr>
<td>IsSupported</td>
<td>Clarifies whether the parameter identified by pParamID is available for the given module or not.</td>
</tr>
<tr>
<td>SetDefState</td>
<td>Resets all of the module parameters to default values.</td>
</tr>
<tr>
<td>EnumParams</td>
<td>Retrieves the list of parameters that are valid for the given module.</td>
</tr>
<tr>
<td>CommitChanges</td>
<td>Verifies and applies the modified parameter values.</td>
</tr>
<tr>
<td>DeclineChanges</td>
<td>Declines all of the parameter modifications that have been made since the last CommitChanges call; sets the module to the previous committed state.</td>
</tr>
<tr>
<td>SaveToRegistry</td>
<td>Saves the committed module state into the registry database.</td>
</tr>
<tr>
<td>LoadFromRegistry</td>
<td>Loads the module parameters from the registry database. The loaded values should be verified and applied by the CommitChanges call.</td>
</tr>
<tr>
<td>RegisterForNotifies</td>
<td>Subscribes the client for the notification messages about the module parameters modification.</td>
</tr>
<tr>
<td>UnregisterFromNotifies</td>
<td>Unsubscribes the client from the notification messages about the module parameters modification.</td>
</tr>
</tbody>
</table>

### ITrimmerObjControl interface

The **ITrimmerObjControl** interface is a specific to the SolveigMM Video Editing Engine. The **ITrimmerObjControl** interface is exposed by the SolveigMM Video Editing Engine and provides methods to start/stop the engine, the rest routines are obsolete. Instead, **IModuleConfig** interface is used.

IID of ITrimmerObjControl interface is
IID _ITrimmerObjControl
{9D9BF96C-8E7A-41FC-ABB1-CCC9CEF493E9}

### Methods in VTable order

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
</table>
**IUnknown**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryInterface</td>
<td>Retrieves pointers to supported interfaces</td>
</tr>
<tr>
<td>AddRef</td>
<td>Increments reference count</td>
</tr>
<tr>
<td>Release</td>
<td>Decrements reference count</td>
</tr>
</tbody>
</table>

**IDispatch**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetTypeInfoCount</td>
<td>Retrieves the number of type information interfaces that an object provides (either 0 or 1)</td>
</tr>
<tr>
<td>GetTypeInfo</td>
<td>Gets the type information for an object</td>
</tr>
<tr>
<td>GetIDsOfNames</td>
<td>Maps a single member and an optional set of argument names to a corresponding set of integer DISPIDs</td>
</tr>
<tr>
<td>Invoke</td>
<td>Provides access to properties and methods exposed by an object</td>
</tr>
</tbody>
</table>

**ITrimmerObjControl**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Starts the engine process</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops the engine process</td>
</tr>
<tr>
<td>put_InputFN</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_InputFile guid</td>
</tr>
<tr>
<td>get_InputFN</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_InputFile guid</td>
</tr>
<tr>
<td>put_OutputFN</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_OutputFile guid</td>
</tr>
<tr>
<td>get_OutputFN</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_OutputFile guid</td>
</tr>
<tr>
<td>get_Status</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_Progress guid</td>
</tr>
<tr>
<td>get_StreamLength</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_Progress guid</td>
</tr>
<tr>
<td>put_StartPosition</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid</td>
</tr>
<tr>
<td>get_StartPosition</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid</td>
</tr>
<tr>
<td>put_StopPosition</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid</td>
</tr>
<tr>
<td>get_StopPosition</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid</td>
</tr>
<tr>
<td>SetCallback</td>
<td>Obsolete. Substituted IModuleConfig interface with SMAT_Callback guid</td>
</tr>
</tbody>
</table>

**ITrimmerObjControl::Start**

The Start method starts the engine with the parameters set

**Syntax**

```c
HRESULT Start (void);
```

**Return Value**

Returns S_OK if the process was successfully started or an HRESULT error code otherwise.

**ITrimmerObjControl::Stop**

The Stop method stops the current engine process

**Syntax**

```c
HRESULT Stop (void);
```

**Return Value**

Returns S_OK if the process was successfully started or an HRESULT error code otherwise.
**ITrimmerObjControlCB interface**

The **ITrimmerObjControlCB** interface is a specific to the SolveigMM Video Editing Engine. The **ITrimmerObjControlCB** interface provides callback methods for an application implemented this interface. The interface is set by **IModuleConfig::SetValue** interface with **SMAT_Callback** parameter guid.

IID of ITrimmerObjControlCB interface is

IID **ITrimmerObjControlCB**

{33968711-8887-46D4-A71B-9B1B061EEDCA}

**Methods in VTable order**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IUnknown</td>
<td></td>
</tr>
<tr>
<td>QueryInterface</td>
<td>Retrieves pointers to supported interfaces</td>
</tr>
<tr>
<td>AddRef</td>
<td>Increments reference count</td>
</tr>
<tr>
<td>Release</td>
<td>Decrements reference count</td>
</tr>
<tr>
<td>ITrimmerObjControlCB</td>
<td></td>
</tr>
<tr>
<td>OnStart</td>
<td>Callback method to indicate the engine started trimming process</td>
</tr>
<tr>
<td>OnStop</td>
<td>Callback method to indicate the engine stopped trimming process</td>
</tr>
<tr>
<td>OnError</td>
<td>Callback method to indicate that some error occurred</td>
</tr>
<tr>
<td>OnPause</td>
<td>Not implemented</td>
</tr>
</tbody>
</table>

**ITrimmerObjControlCB::OnStart**

The OnStart callback method is called indicating the engine started trimming process

**Syntax**

```c
HRESULT OnStart (void);
```

**Return Value**

Returns S_OK if successful, or an HRESULT error code otherwise.

**ITrimmerObjControlCB::OnStop**

The OnStop callback method is called indicating the engine stopped trimming process

**Syntax**

```c
HRESULT OnStop (void);
```

**Return Value**

Returns S_OK if successful, or an HRESULT error code otherwise.

**ITrimmerObjControlCB::OnError**

The OnError callback method is called indicating some error occurred

**Syntax**

```c
HRESULT OnError (HRESULT hr, BSTR Val);
```
Parameters
hr – Variable to contain error code
Val – Variable to contain error description

Return Value
Returns S_OK if successful, or an HRESULT error code otherwise

SolveigMM Video Editing Engine Parameters

The following section describes the Parameter GUIDs declared in the PropID_EditingEngine.h header file to be used for tuning the SolveigMM Video Editing Engine filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs for the SolveigMM Video Editing Engine.

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Available values</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_InputFile</td>
<td>VT_BSTR</td>
<td>-</td>
<td>-</td>
<td>Specifies an input media file name</td>
</tr>
<tr>
<td>SMAT_OutputFile</td>
<td>VT_BSTR</td>
<td>-</td>
<td>-</td>
<td>Specifies an output media file name</td>
</tr>
<tr>
<td>SMAT_Flags</td>
<td>VT_UI4</td>
<td>[SMM_TrimFlags]</td>
<td>0</td>
<td>Sets trimming configuration flags to be set</td>
</tr>
<tr>
<td>SMAT_ResetFlags</td>
<td>VT_UI4</td>
<td>[SMM_TrimFlags]</td>
<td></td>
<td>Sets trimming configuration flags to be reset</td>
</tr>
<tr>
<td>SMAT_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
<td>0</td>
<td>Sets trimming parameters via TrimInfoList structure</td>
</tr>
<tr>
<td>SMAT_Progress</td>
<td>VT_I4</td>
<td>[0 - 100] Read-only</td>
<td>-</td>
<td>Retrieves a progress of trimming process</td>
</tr>
<tr>
<td>SMAT_AudioStreamNumber</td>
<td>VT_I4</td>
<td>[1 - n]</td>
<td>1</td>
<td>Sets a number of an audio stream to be trimmed</td>
</tr>
<tr>
<td>SMAT_VideoStreamNumber</td>
<td>VT_I4</td>
<td>[1 - n]</td>
<td>1</td>
<td>Sets a number of a video stream to be trimmed</td>
</tr>
<tr>
<td>SMAT_Callback</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
<td>-</td>
<td>Sets a callback COM interface: ITrimmerObjControlCB</td>
</tr>
<tr>
<td>SMAT_GetVersions2</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
<td>Read only</td>
<td>Retrieves all the objects versions being used by the engine</td>
</tr>
<tr>
<td>SMAT_CurrentSize</td>
<td>VT_UI8</td>
<td>Read only</td>
<td>-</td>
<td>Retrieves output AVI file size during trimming process</td>
</tr>
<tr>
<td>SMAT_BatchFileName</td>
<td>VT_BSTR</td>
<td>-</td>
<td>-</td>
<td>Specifies an input batch file name</td>
</tr>
<tr>
<td>SMAT_CurrentBatchTask</td>
<td>VT_UI4</td>
<td>Read only</td>
<td>-</td>
<td>Retrieves a number of a current task being processed</td>
</tr>
<tr>
<td>SMAT_OutputType</td>
<td>VT_UI4</td>
<td>[SMM_OutputTypes] input type</td>
<td></td>
<td>Specifies an output file type</td>
</tr>
<tr>
<td>SMAT_BatchTasksNum</td>
<td>VT_UI4</td>
<td>Read only</td>
<td>-</td>
<td>Retrieves a count of tasks a set batch file contains</td>
</tr>
<tr>
<td>SMAT_TaskType</td>
<td>VT_UI4</td>
<td>[SMM_TaskType]</td>
<td>0</td>
<td>The type of process to be performed</td>
</tr>
<tr>
<td>SMAT_TrackInfo</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
<td>-</td>
<td>Passes the parameters of the chosen files to Editing Engine</td>
</tr>
</tbody>
</table>
### SMAT_MPEG2IndexFileName

**Type:** VT_BSTR  
**GUID:**  
{85x809}  
**Description:**  
Provides the name of MPEG2 index filename.

### SMAT_SilentPath

**Type:** VT_BSTR  
**GUID:**  
{85x809}  
**Description:**  
Sets the registry path that stores paths of filters to use in silent mode.

### SMAT_MPEG2CodecsSet

**Type:** VT_BSTR  
**GUID:**  
{85x809}  
**Description:**  
Sets the set of codecs to use for MPEG2 files editing. Should be replaced with SMAT_CodecsSet.

### SMAT_CodecsSet

**Type:** VT_BSTR  
**GUID:**  
{85x809}  
**Description:**  
Sets the set of codecs to use for files editing.

### SMAT_GetFileType

**Type:** VT_BSTR  
**GUID:**  
{85x809}  
**Description:**  
Sets the set of codecs to use for files editing.

### SMAT_ValidateFiles

**Type:** VT_I4  
**GUID:**  
{85x809}  
**Description:**  
Validate the parameters of chosen files.

### SMAT_ASFMarker

**Type:** VT_BYREF  
**GUID:**  
{85x809}  
**Description:**  
Sets or retrieves ASF marker with specified name and position.

### SMAT_ASFMarkersCount

**Type:** VT_I4  
**GUID:**  
{85x809}  
**Description:**  
Retrieves a count of markers contained in an input ASF.

### SMAT_ASFMarkerRemove

**Type:** VT_BSTR, VT_I4  
**GUID:**  
{85x809}  
**Description:**  
Removes a marker from an input ASF.

---

**SMAT_InputFile**

**GUID:**  
{4890910-6B0B-465D-B507-2568710565FD}  
**Description:**  
Specifies an input media file name (AVI, WMV, ASF, MP3, MPA, WMA). For details please see SMM Trim C++ AVI, MP3, BatchSplit sample applications.

**Type:**  
VT_BSTR

---

**SMAT_OutputFile**

**GUID:**  
{45c2624-07e5-4905-A1A5-23DDD26EFC13}  
**Description:**  
Specifies an output media file name the Engine saves processed data to. For details please see SMM Trim C++ AVI, MP3, BatchSplit sample applications.

**Type:**  
VT_BSTR

---

**SMAT_FLAGS**

**GUID:**  
{AAF40DFF-0698-4424-87DC-3B1E2D60CDD8}  
**Description:**
Set trimming configuration flags to the Video Editing Engine. For details please see SMM Trim C++ MP3 sample application.

**Type:**
VT_UI4

**Note:**
Value 0 means reset all flags.

### SMAT_ResetFlags

**GUID:**
{19F90CB9-A5EA-4821-A8BA-AD6FB9422B9C}

**Description:**
Resets specified flags that were set before by means of SMAT_Flags parameter. For details please see SMM Trim C++ MP3 sample application.

**Type:**
VT_UI4

### SMAT_TrimList

**GUID:**
{C1D0F958-4489-4322-80CD-898089A220C1}

**Description:**
Specifies trimming parameters to be a pointer to TrimInfoList structure that contains start/stop time values of media fragments to kept. Time values are 100 nanosecond units. For details please see SMM Trim C++ AVI, MP3 sample applications.

**Type:**
VT_BYREF | VT_UI1

### SMAT_Progress

**GUID:**
{AE82DE01-D84B-4EE5-90AA-052B108C4380}

**Description:**
Retrieves a progress of trimming process in percentage. For details please see SMM Trim C++ AVI, MP3, sample applications.

**Type:**
VT_I4

**Available Values:**
0 – 100 (%)
Type: VT_I4

Available Values:
0 – do not pass an audio stream to output
[1 - n] – number of an audio stream, an input media files contains, that must be trimmed and presented in output file

SMAT_VideoStreamNumber

GUID:
{D9898809-5987-4EBC-967D-C595465923A1}

Description:
Sets a number of a video stream to be trimmed

Type: VT_I4

Available Values:
0 – do not pass a video stream to output
[1 - n] – number of a video stream, an input media files contains, that must be trimmed and presented in output file

SMAT_Callback

GUID:
{DA1ED33F-9EF4-4c98-BCA5-66FF917FFCBE}

Description:
Sets a callback COM interface – ITrimmerObjControlCB. For details please see example code and BatchSplit, SMM Trim C++ AVI sample applications.

Example Code:
VARIANT var;
HRESULT hr = S_OK;
var.vt = (VT_BYREF | VT_UI1);
var.pbVal = (BYTE*) &m_pITrimmerObjControlCB;
hr = m_pIModuleConfig->SetValue(&SMAT_Callback, &var);

Type: VT_BYREF | VT_UI1

SMAT_GetVersions2

GUID:
{DB90789E-8C00-4d40-B542-428C8DC20169}

Description:
Retrieves all the objects names and versions being used by the engine. The data is stored to FILE_VERSIONS_2 structure a pointer to that is an input parameter. Memory for FILE_VERSIONS_2::data has to be allocated by an application. First application asks the Video Editing object count of objects by setting NULL to input parameter. For details please see an example code.

Available Values:
NULL – means the engine has to retrieve a byte size of structure

Example Code:
VARIANT var;
var.vt = (VT_BYREF | VT_UI1);
var.pbVal = NULL;
HRESULT hr = m_pIModuleConfig->GetValue( &SMAT_GetVersions2, &var );
FILE_VERSIONS_2 list;
list.count = var.llVal;
list.data = new VERSION_INFO_2[list.count];
var.vt  = ( VT_BYREF | VT_UI1 );
var.pbVal = ( BYTE * )&list;
hhr = m_pIModuleConfig->GetValue( &SMAT_GetVersions2, &var );

**SMAT_CurrentSize**

**GUID:**
{BD62FD5E-B517-4f8f-976E-5FA1ACE4F9DC}

**Description:**
Retrieves an output AVI file byte size during trimming process.

**Type:**
VT_UI8

**SMAT_BatchFileName**

**GUID:**
{909605C1-F9CE-4360-9AFA-0BAE16E95784}

**Description:**
Specifies an input batch file name (*.xtl). In this mode the Video Editing Engine process a batch file set. For details see BatchSplit sample application.

**Type:**
VT_BSTR

**SMAT_CurrentBatchTask**

**GUID:**
{7AF7B8FE-FA2F-44f2-915E-D668EE32E8A6}

**Description:**
Retrieves a number of a current task being processed by the Video Editing Engine in batch mode. For details see BatchSplit sample application.

**Type:**
VT_UI4

**SMAT_OutputType**

**GUID:**
{3F20F1D6-1132-4e28-AB6B-ED94700C2949}

**Description:**
Specifies output file type as SMM_OutputTypes parameter

**Type:**
VT_UI4
Available Values:
For details see SMM_OutputTypes available values

**SMAT_BatchTasksNum**

**GUID:**
{909605C1-F9CE-4360-9AFA-0BAE16E95784}

**Description:**
Retrieves count of tasks a batch file set contains. For details see BatchSplit sample application.

**Type:**
VT_UI4

**SMAT_TaskType**

**GUID:**
{A5E6000D-52D9-4f41-9091-11C4FDAB990B}

**Type:**
VT_UI4

**Description:**
Specifies which process is to be performed with chosen files (See SMM_TaskType).

**Available values:**
see Editing Engine enumerations and structures.

**SMAT_TrackInfo**

**GUID:**
{FA8BD847-4DC6-492d-9585-E86654CDAC1F}

**Type:**
VT_BYREF | VT_UI1

**Description:**
Passes the parameters of the chosen files to Editing Engine. Used for joining, the application should correctly fill in the SMM_TRACK_INFO's members and pass the pointer to the class to Editing Engine via IModuleConfig interface (for more information see Editing Engine enumerations and structures, Editing Engine parameters).

**SMAT_MPEG2IndexFileName**

**GUID:**
{1ADE47B6-BB44-4820-A077-6116F861775B}

**Type:**
VT_BSTR

**Description:**
Sets the MPEG2 index file name to achieve maximum trimming precision.

**SMAT_SilentPath**

**GUID:**
{2E83B26B-65B3-4609-95B5-E918A96AE151}
Type:
VT_BSTR

Description:
Sets the registry path that stores paths of filters to use in silent mode (without registration).

**SMAT_MPEG2CodecsSet**

GUID:
{534D4154-5F4D-5045-4732-436F666967}

Type:
VT_BSTR

Description:
Sets the vendor of codecs to use for MPEG2 files editing. The former versions of SDK used to support MainConcept codecs (MC). Current uses Elecard (EL). Used for compatibility should be replaced with SMAT_CodecsSet.

Default value:
dmx_EL__mux_EL__enc_EL__dec_EL__ixr_EL__ixw_EL. The current version of SDK uses Elecard filters to edit MPEG2 files.

**SMAT_CodecsSet**

GUID:
{17B25796-46AB-44a6-BBF4-9A3A160E63FB}

Type:
VT_BSTR

Description:
Sets the profile that shows which filter should be used for current operation.

Default value:
dmx<type:MPEG2>_EL__mux<type:MPEG2>_EL__enc<type:AVC>_EL__enc<type:MPEG2>_EL__dec<type:AVC>_EL__dec<type:MPEG2>_EL__ixr<type:MPEG2>_EL__dmx<type:MP4>_SM__mux<type:MP4>_SM.
The current version of SDK uses Elecard muxer/splitter/encoder/decoder/index reader filters to edit MPEG2 and AVCHD files and Solveig Multimedia muxer/demuxer for MP4 files.

**SMAT_GetFileType**

GUID:
{534D4154-5F47-6574-4669-6C6554797065}

Type:
Output parameter - VT_INT(intVal) - File type (should be casted to SMM_OutputTypes)

Description:
this function returns the type of the file. SMM_File_Type_NO means that the type is not supported or it is non-media file. For more information see SMM_OutputTypes available values.

Example Code:

```c
SMM_OutputTypes FileType = SMM_File_Type_NO;
VARIANT var = {0};
HRESULT hr = S_OK;
```
CString strFileName = /* Some file's name */
var.vt = VT_BSTR;
Var.bstrVal = strFileName.AllocSysString();
hr = m_pIModuleConfig->SetValue( &SMAT_InputFile, &var );
hr = m_pIModuleConfig->GetValue( &SMAT_GetFileType, &var );
if( FAILED(hr) )
{
    // Place here the appropriate failure handler
}
FileType = ( SMM_OutputTypes ) var.intVal;

GUID:
{534D4154-5661-6C69-6461-746546696C65}

Type:
Input - not applied;
Output - VT_I4(Val);

Description:
This parameters can be applied only to the joining process using IModuleConfig interface. This is used to validate the parameters of chosen files. Files might have different parameters, and joining filter doesn't support reencoding. It means that before joining you should validate if the files are compatible. IModuleConfig::GetValue should be used.
Three results are possible:

- the files have equal parameters - Success, the joining can be started;
- the files have different secondary parameters (mostly applied for sound stream) - the joining still can be started, but there might be problems while playing an output file;
- the files have different critical parameters - the files are incompatible and the joining is impossible. The critical parameters - file type, video and audio compression, presence/absence of video/audio stream, video frame height and width.

Example Code:
VARIANT var = {0};
HRESULT hr = S_OK;
h = m_pIModuleConfig->GetValue( &SMAT_ValidateFiles, &var );
if( FAILED(hr) )
{
    HRESULT hrValidationRes = (HRESULT) var.lVal;
    if( SUCCEEDED(hrValidationRes) )
    {
        //Check if hrValidationRes == S_FALSE(see further)
        //and use an appropriate handler{ if needed }
        //then start SMM Editing Engine
    }
    else
    {
        //Place here the appropriate error handler{ if any }
        //and return without starting
    }
}

For more information on how to correctly tune SMM Editing Engine for joining, see the SMM Join C++ Sample source code and pay attention on the comments.

Available values:
The output parameter is a HRESULT of the files validation. The most important possible values are as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_INVALIDARG</td>
<td>Type of the output file or structure that describes TrackInfo wasn't set.</td>
</tr>
<tr>
<td>VFW_E_INVALID_FILE_FORMAT</td>
<td>Some files have invalid or unsupported format.</td>
</tr>
<tr>
<td>S_OK</td>
<td>SUCCESS, files can be joined.</td>
</tr>
<tr>
<td>S_FALSE</td>
<td>The files have different secondary parameters, it may cause problems.</td>
</tr>
<tr>
<td>E_FAIL</td>
<td>Some critical parameters are different. Further joining is impossible and must be stopped here.</td>
</tr>
</tbody>
</table>

Notes:
Files validation should be performed before starting the SMM Editing Engine (See ITrimmerObjControl::Start).
To check if the files are compatible, cast the result's lVal to HRESULT.
If you check the result with "SUCCEEDED()" macros remember that it returns TRUE if HRESULT( lVal ) == S_FALSE and FALSE if HRESULT( lVal ) == E_FAIL or other errors. You can avoid using this automatic validation due to abscense of intermediate results that makes impossible to locate the incompatible files. Instead you may use your own, but in this case you should consider all the notes and features described above. method.

Required steps before start joining
To perform the joining task correctly, you should make the following:

- **Initialize the SMM Editing Engine.** If you use the only instance of the editing engine, you can initialize it once when the application starts. You can optionally set the ITrimmerObjectCB interface to the editing engine (See ITrimmerObjControlCB interface and SMAT_Callback for more information).
  This step is shown in the InitEngine() member function of SimpleJoiner Sample.
- **Make the list of files to be joined**, and make sure they have the same type. **It is important step.** To do this, use SMAT_GetFileType parameter (See other Editing Engine Parameters for more information).
- Initialize the SMM_GROUP_TASK and create the appropriate number of clips. Then fill in allocated SMM_CLIP_INFO and SMM_GROUP_TASK (See other Editing Engine structures) with the most significant parameters.
  You should fill structures as follows:
  - **szDstName** (SMM_GROUP_TASK) - the output file's name;
  - **nTaskType** (SMM_GROUP_TASK) - should be set to SMM_Task_Type_Joining;
  - **tracks** (SMM_GROUP_TASK) - contains the pClips structure. Only the tracks[0] should be used.
  - **nVideoStream / nAudioStream** (SMM_TRACK_INFO) - the ordered number of video/audio stream to be used for joining. Because files may contain multiple streams these both parameters should be set to -1, which means that all streams in the file would be used;
  - **pClips** (SMM_TRACK_INFO) - pointer to an array of clips. The array elements' number should be the same as the number of parts to be joined (currently equal to the count of files). Each element should describe single part to be joined (the whole file or separate interval of a file).
  - **dwClipNum** - ordered number of the clip;
  - **rtStart / rtEnd** - the start/endpoint of the interval of file to be joined. Currently these fields should be set to 0, which means that the whole file should be appended to the output.
  - **wsfName** - the name of the file to be appended.

  This step is shown in the GetGroupTaskFromList() member function.
- **Set the following parameters** of SMM Editing Engine:
  - SMAT_TaskType,
  - SMAT_OutputFile,
  - SMAT_TrackInfo,
See Editing Engine Parameters for more information. Don't forget to free memory allocated for the SMM_TRACK_INFO member of SMM_GROUP_TASK. Also, please note that the settings won't take place until they will be committed due to IModuleConfig specification. (See IModuleConfig::CommitChanges).

- **Validate files compatibility** (for more information see SMAT.ValidateFiles parameter description).

After you've performed all this steps and if the validation succeeded, the SMM Editing Engine is fully tuned up for joining. You can start joining now.

All these steps are shown in SimpleJoiner Sample source codes (mostly in ConfigureTrimmerObject() member function). Please, pay attention on the comments.

---

**SMAT_ASFMarker**

**GUID:**
{C022A53F-CE3D-4c03-BAED-11FCC7CCDED2}

**Description:**
Sets or retrieves ASF marker with specified name and position to an input file. An input parameter is a pointer to SMM_ASF_MARKER structure. For details please see ASFMarkers sample application.

**Type:**
VT_BYREF

**Available Values:**
Retrieving ASF marker data is possible by specifying a marker name (see SMM_ASF_MARKER::name) or a marker index (SMM_ASF_MARKER::num_pos)

**SMAT_ASFMarkersCount**

**GUID:**
{0B9C6AB5-F968-46A8-85E1-09FA8B8A734B}

**Description:**
Retrieves a count of markers contained in an input ASF file

**Type:**
VT_I4

**SMAT_ASFMarkerRemove**

**GUID:**
{72825BB9-B725-456A-8E2C-D650ACDFB674}

**Description:**
Removes a marker from an input ASF file by marker name or its index. An input parameter can be either marker name (VT_BSTR) or its zero-based index (VT_I4)

**Type:**
VT_BSTR, VT_I4
SolveigMM BMP to AVI

SMM BMP to AVI is a set of utilities like DLL, streaming DS filter, SolveigMM BMP to AVI .NET application.

The DirectShow filter is used to produce the video stream from input BMP pictures. Visual C++ DLL provides the necessary functions to operate with streaming filter using it to create AVI file. Visual C# application is to demonstrate the use of DLL API.

An available functions are muxing BMPs into the AVI video stream with the given frame duration. The output stream can be compressed with one of the available encoders or uncompressed (RGB24). DLL also allows getting information from AVIs, like the frame sizes, compression method, the duration of a frame and total number of frames in the given AVI. Also DLL can be used to get the screen shot (24bit BMP) of the exact frame from existing AVI file.

24-32bits BMP pictures are supported as an input (in 32-bit BMP A-bit is ignored).

BmpToAvi.dll

The BmpToAvi.dll provides methods to create the AVI file from different BMP pictures, read the information from the AVI file, and get the exact frame from AVI stream as a bitmap.

Requirements

The dll operates with 24 and 32 bit BMPs to create AVI. For 32 bits BMP the A channel is ignored due to the output format (MEDIASUBTYPE_RGB24).

The dll doesn't support resizing, which means all BMPs must have the same sizes (width and height). The picture with different sizes would be skipped.

Features

The dll provides the following set of functions:

CreateNewStream;
ReleaseStream;
OpenAVIStream;
ChooseEncoder;
SetStreamProperties;
WriteImage;
CloseAVIStream;
GetAVIFileInfo;
ReadImageFromStream;
FreeImage;

- **CreateNewStream** – the basic operation of stream creating. All the following operations are applied to the created stream.
- **OpenAVIStream** – initiates the appropriate graph, depending on the specified name of the file. If the existing file is specified, this file would be opened in read only mode. This mode allows to read the properties of the specified AVI, or get specified frame from the stream as a 24-bit bitmap. If the new file is specified, this file would be opened in file creating mode. This mode allows to write the BMP into avi file and also get the properties of the file being created, but the reading bitmap from stream is not allowed.
- **ChooseEncoder** – the function must be used if the file is not exist, before the stream is opened and mustn't be used with the existing AVI files. It lets you choose the desired encoder to compress the stream with the exact format. Please make sure that not all encoders are compatible with AVI container. By default – no encoders are chosen, the stream is uncompressed.
**SetStreamProperties** – the function must be used if the file is not exist, before the stream is opened and mustn’t be used with the existing AVI files. It sets the basic parameters of output AVI stream: Width, Height, Frame Duration. If this parameters aren’t set the AVI stream couldn’t be opened.

**WriteImage** – writes the specified image into the stream opened in file creating mode.

**GetAVIFileInfo** – reads properties from the opened stream. The properties to be read are: frame height, frame width, frame duration, stream compression, total number of frames in the stream.

**CloseAVIStream** – releases the graph created by OpenAVIStream. Has different behavior for file creating mode depending on passed parameter: normal – all bitmaps are passed in and the stream is only closed after all passed bitmaps are written into the stream; emergency – the stream is closed right when the function is called, all queued bitmaps would be rejected.

**ReadImageFromStream** – only available in read only streams – gets the specified frame from stream as a buffer containing 24-bit bitmap data array.

**FreeImage** – destroys the buffer created by ReadImageFromStream. Use this function to avoid memory leaks.

**ReleaseStream** – destroys the stream created by CreateNewStream. Use this function when the stream is to be useless. Frees all resources and memory grabbed during the current stream work.

**Usage**

To create file:

- Create the stream with **CreateNewStream**. The returned value is the unique identifier of the stream, you must keep it for further use. All other functions require this identifier.
- Set the parameters of the stream with **SetStreamProperties** and optionally set the encoder with **ChooseEncoder**.
- Start streaming with **OpenAVIStream**. Now the AVI stream is created and images can be written.
- Use the **WriteImage** to write the necessary bitmaps into AVI stream.
- When all bitmaps are written use **CloseAVIStream** to release the graph. The created avi file could be opened with other applications now.
- If the stream isn’t going to be used for other needs destroy it with **ReleaseStream**.

To operate with more than one AVI at once it is recommended to create a separate thread for each AVI and create the stream inside this thread. To control the creation progress the **GetAVIFileInfo** can be used to find out the number of frames currently been written into the stream. Use the emergency type of **CloseAVIStream** (CloseAVIStream(TRUE)) to terminate or cancel the process.

Existing file:

- Create the stream with **CreateNewStream**. The returned value is the unique identifier of the stream, you must keep it for further use. All other functions require this identifier.
- Open the existing file in read only mode with **OpenAVIStream**.
- Use the **ReadImageFromStream** to get a screen shot of the current frame from stream or **GetAVIFileInfo** to read the AVI stream properties.
- If you used **ReadImageFromStream**, release the received buffer with **FreeImage** when you’ve done with it.
- If you’ve finished with the stream release the graph with **CloseAVIStream**.
- Destroy the stream with **ReleaseStream**. You can skip the **CloseAVIStream** action and use **ReleaseStream** only.

---

**SolveigMM BMP Push Source Filter**

The SMM_BMPPushSrc.ax is a streaming DirectShow filter that produces the video stream from input bitmaps. Stream parameters are: MEDIATYPE_Video, MEDIASUBTYPE_RGB24, FORMAT_VideoInfo.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IFileSourceFilter, IModuleConfig, IAMFilterMiscFlags, ISpecifyPropertyPages</th>
</tr>
</thead>
</table>
SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Input Pin Media Types</th>
<th>Not applicable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Interfaces</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Output Pin Media</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_RGB24</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_BMPPushSource</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>CLSID_BMPPushSource_Props</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_BMPPushSrc.Ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_BMPPushSource.h</td>
</tr>
</tbody>
</table>

**Requirements**

The filter operates with 24 and 32 bit BMPs as an input. For 32 bits BMP the A channel is ignored due to the output format (MEDIASUBTYPE_RGB24). The filter doesn’t support resizing, which means all BMPs must have the same sizes (width and height). The picture with different sizes would be skipped.

**Features**

The filter can be tuned up programmatically or manually via the property page (e.g. using Microsoft Graph Editor). The property page (fig. 1) includes the following control elements:

- Table containing the list BMP files,
- **Add Files** button – opens the “Open File” dialog to choose the desired files (alternatively drag-n-drop can be used).
- **Delete Chosen** – removes the selected files from list.
- **Loop list** – if checked, the list would be repeated from the beginning after the last file will be sent until the graph stopped externally.
- **Frame length** – the DirectShow Reference time value (100E-9sec) specifying the frame duration (frame rate). The default value 400000 is equal to 40msec which is corresponded to 25 Hz( fps ).
- **Width** and **Height** – information fields. Their values are set according to the the first frame in the list and would be used to check if other BMPs could be appended into stream. If some BMP has different sizes they would be skipped.

![Property Page Screenshot]

**Usage**

NOTE: All changes must be applied (using APPLY button in Graph Editor PropertyPage dialog or CommitChanges method of IModuleConfig interface) to take place, the default values are used otherwise.

To tune the filter up manually:
- Add filter to a graph.
- Add some BMPs into the list.
- Type the desired frame length.

Now the filter can be connected to other downstream filters.
- If you want to repeat current list, check the “Loop List”.
- Build Graph.
- After all files from the list would be passed downstream the graph would be stopped automatically, however if the “Loop list” is specified you must stop the graph manually.

To tune up programmatically not using property page:
In case the property page is not accessible or you don’t want to use it, there is a set of parameters, that can be used via IModuleConfig interface.

Here are the existing parameters and their purposes:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMBPS_StrmProps</td>
<td>VT_BYREF</td>
<td>Set/retrieve the stream parameters</td>
</tr>
<tr>
<td>SMMBPS_FilesTabl</td>
<td>VT_BYREF</td>
<td>Set/retrieve the list of files</td>
</tr>
<tr>
<td>SMMBPS_LoopList</td>
<td>VT_BOOL</td>
<td>BMP to be appended</td>
</tr>
<tr>
<td>SMMBPS_IsQueue</td>
<td>VT_BOOL</td>
<td>Checks if the filter’s internal samples queue is empty</td>
</tr>
<tr>
<td>SMMBPS_Terminate</td>
<td>VT_BOOL (boolVal)</td>
<td>Terminates the stream creation and stops graph</td>
</tr>
</tbody>
</table>

**SMMBPS_StrmProps**

**GUID:**
{534D4D42-5F53-74726D50726F7073}

**Description:**
Duplicates the property page control, used to set/retrieve the stream parameters. Uses the StreamParameters structure (see the implementation of the structure below).

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMMBPS_FilesTabl**

**GUID:**
{534D4D42-5F46-696C6573546126C}

**Description:**
Duplicates the property page controls, used to set/retrieve the list of files.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMMBPS_LoopList**

**GUID:**
{534D4D42-5053-5F4C-6F6F704C69737400}

**Description:**
Duplicates the property page control.

**Type:**
VT_BOOL (boolVal)
**SMMBPS_InputBMP**

**GUID:**
{534D4D42-5053-5F49-6E707574424D5000}

**Description:**
Passes the BMP to be appended into the stream as a HBITMAP. The HBITMAP handle should be casted to BYTE*

**Type:**
not applied (pvRecord)

**SMMBPS_IsQueue**

**GUID:**
{534D4D42-5053-5F49-7351756575650000}

**Description:**
Checks if the filter’s internal samples queue is empty.

**Type:**
VT_BOOL (boolVal)

**SMMBPS_Terminate**

**GUID:**
{534D4D42-5053-5F54-65726D696E617465}

**Description:**
Terminates the stream generation and stops graph. Need no extra parameters passed.

**Type:**
not applied

Here is the definition of `StreamParameters` structure

```c
typedef struct StreamParams {
    LONG lWidth; //Frame width
    LONG lHeight; //Frame height
    REFERENCE_TIME rtFrameLenght; //One frame duration
    LONGLONG llTotalFrames; //Total number of frames
    DWORD dwCompression; //Used compression
}StreamParameters;
```

These parameters allow two types of the filter usage.
1) The same as described above. The only difference is that you must provide valid Width and Height of a first BMP using the SMMBPS_StrmProps.

2) Produce video stream using SMMBPS_InputBMP parameter. In this case you should do the following:
   - Add filter to the graph and obtain the IModuleConfig interface.
   - Set valid parameters of the stream using StreamParameters structure (only Width, Height and FrameLength are valuable). After that it is possible to connect the filter with the downstream.
   - Pass all desired bitmaps as a HBITMAP (can be obtained with WIN GDI/GDI+) one by one, using SMMBPS_InputBMP.
   - After all bitmaps are passed, use the SMMBPS_IsQueue to find out if the filter passed the bitmaps downstream, because the filter has an internal queue so it might take some time to pass them all.
   - When you are sure that all bitmaps are passed, stop the graph.
SolveigMM Video Editing SDK

- You can also use SMMBPS_Terminate. It will stop the graph and release all internal resources. It can be used to implement canceling the operation, in this case you don’t need to use SMMBPS_IsQueue, and all samples that are already in the internal queue but not yet delivered downstream will be released.

**SolveigMM Media Joiner Filter**

The SMM_MediaJointer.ax is a DirectShow dump filter. It lets you join the fragments of video or audio streams without re encoding.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IFileSinkFilter, IModuleConfig, IAMFilterMiscFlags, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Audio, MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_Jointer</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>CLSID_SMM_Jointer_Prop_Main</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_MediaJointer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_MediaJoiner.h</td>
</tr>
</tbody>
</table>

**Requirements**

The filter currently supports the following formats: AVI, Windows Media files (ASF, WMV, WMA), MPEG1 Audio (Layer 1-3), WAV, MP4, MOV, MKV, MPEG-1, MPEG-2, AVCHD. The files to be joined must have compatible parameters (See SMAT_ValidateFiles for more information), usually they should have the same type, compression, widths/heights.

**Features**

The filter has the only parameter to tune up which is "Output type". The property page is shown on fig. 1) and includes the following control elements:

- Table containing the refreshable joining statistic of each pin.
- Output file name
- Output Type combo-box – chooses the type of file (currently available types are AVI, ASF, MPA; ASF includes all Windows media files, MPA includes all MPEG1 audio).
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph and specify the output file name.
- Set the correct file type.

Now the filter can be connected to other filters.

- Build Graph and start it.
- When the graph is stopped remove the source filter and add another file you’d like to be appended. Never remove the Media Joiner Filter unless you have no files to join.
- Build Graph and start it again. Repeat it with all files to be appended.
- When all files are joined, just remove all filters from the graph.

To use programmatically not using property page:

If it is necessary to control the filter from inside the external application there is a set of parameters to access the filter’s properties through the IModuleConfig.

Here are the existing parameters and their purposes:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMJ_FileName</td>
<td>VT_BSTR</td>
<td>Set/retrieve the output file's name</td>
</tr>
<tr>
<td>SMMJ_BytesWrittenByPin</td>
<td>in - VT_INT, out - VT_UI8</td>
<td>Retrieve the number of bytes written by specified pin</td>
</tr>
<tr>
<td>SMMJ_GetFullStatistic</td>
<td>VT_BYREF</td>
<td>Retrieves the full writing statistic of each pin. (See below)</td>
</tr>
<tr>
<td>SMMJ_StopWriting</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SMMJ_FileType</td>
<td>VT_INT</td>
<td>Set/retrieve the output file's type.</td>
</tr>
<tr>
<td>SMMJ_AvgTimeWritten</td>
<td>VT_I8</td>
<td>Retrive the current duration of the output file.</td>
</tr>
<tr>
<td>SMMJ_ExternalMessageWindow</td>
<td>-</td>
<td>Set the external window to receive the graph messages.</td>
</tr>
</tbody>
</table>

**SMMJ_FileName**

**GUID:**
{534d4d4a-4f49-4e54-4552-464C4E414d45}

**Description:**
Used to set/retrieve the output file's name.

**Type:**
VT_BSTR (bstrVal)
SMMJ_BytesWrittenByPin

**GUID:**
{534dd4d4a-4f49-4e54-4552-234259544553}

**Description:**
Used to retrieve the number of bytes written by the specified pin. Use the number of pin as an input parameter.

**Type:**
in - VT_INT (intVal)
out - VT_UI8 (ullVal)

**Example:**

```c
VARIANT var = {0};
var.intVal = 2; //Pin #2
pModuleConfig->GetValue( &SMMJ_BytesWrittenByPin, &var );
ULLONGLONG ullBytes = var.ullVal;
```

SMMJ_FileType

**GUID:**
{534dd4d4a-4f49-4e54-4552-464C54595045}

**Description:**
Set/Retrieve the type of the output file (member of the SMM_OutputTypes enumeration). If this parameter isn't set the graph wouldn't start.

**Type:**
VT_INT (intVal)

SMMJ_AvgTimeWritten

**GUID:**
{534dd4d4a-4f49-4e54-4552-41564754494D}

**Description:**
Retrieve the duration of currently written output file. Information purpose only.

**Type:**
VT_I8 (llVal)

SMMJ_ExternalMessageWindow
GUID: {534d4d4a-4f49-4e54-4552-4D4553574E44}

Description:
Specifies the external window to receive Graph messages.

Type:
not applied (pvRecord)

SMMJ_GetFullStatistic

GUID: {534d4d4a-4f49-4e54-4552-475453544154}

Description:
Retrieve the full writing statistic of each connected pin. Allocate the array of StatParams (See below). Consider allocating enough memory to receive all pins parameters (filter can have less then 20 pins). Do not forget to free the allocated memory.

Type:
VT_BYREF | VT_UI1 (pbVal)

Here is the definition of StatParams structure

```c
typedef struct StatisticParameters {
    int TotalPins;
    int PinNum;
    LONGLONG llFramesCount;
    ULONGLONG ullBytesCount;
    BOOL EOS;
    GUID Mediatype;
} StatParams;
```

Members:

**TotalPins**
Total number of pins;

**PinNum**
Current pin's number

**llFramesCount**
Frames written by current pin

**ullBytesCount**
Bytes written by current pin

**EOS**
Has the pin already received end of stream notification
Mediatype
Major type of the current pin

SolveigMM MKV Muxer filter

The MKV/WEBM muxer filter implemented as a DirectShow filter. Its purpose is to multiplex input video/audio/subtitles into Matroska or WEB Matroska containers.

Filter Interfaces
IBaseFilter, IModuleConfig, ISpecifyPropertyPages

Input Pin Media Types
MEDIATYPE_Video, MEDIASUBTYPE_NULL
MEDIATYPE_Audio, MEDIASUBTYPE_NULL

Input Pin Interfaces
IPin, IMemInputPin

Output Pin Media Types
MEDIATYPE_Stream, MEDIASUBTYPE_NULL

Output Pin Interfaces
IPin, IMemInputPin

Filter CLSID
CLSID_SMM_MKV_MUXER

Property Page CLSID
Properties - {31741855-F982-40c0-A456-4C8BD7D132A7},
About - {222F030C-2F18-48e8-A8A1-D72FE1C89513}

Executable
SMM_MKVMuxer.ax

Merit
MERIT_DO_NOT_USE

Filter Category
CLSID_LegacyAmFilterCategory

Declared in
PropID_MatroskaMuxer.h

Features

The filter supports MKV and WEBM as an output formats.
The filter's property page is a table shown on fig. 1. It includes the following control elements:
- DocType - output stream type (matroska or webm).
- A set of properties for each input stream (including Track's name and language and also enabling/disabling current track).
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_MatroskaMuxer.h header file to be used for tuning the SolveigMM Matroska Muxer Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Matroska Muxer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMKVM_MUXTIME</td>
<td>VT_I8</td>
<td>Reads the current muxing time</td>
</tr>
<tr>
<td>SMMKVM_TRACKSCOUNT</td>
<td>VT_I2</td>
<td>Reads a number of input streams.</td>
</tr>
<tr>
<td>SMMKV_DOCTYPE</td>
<td>VT_I4</td>
<td>Specifies an output stream type.</td>
</tr>
<tr>
<td>SMMKVM_TRACKPROPS</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
</tbody>
</table>

**SMMKVM_MUXTIME**

**GUID:**
{08AE82FB-A9CB-442c-A087-9C59FD2D7E07}

**Description:**
Used to retrieve current muxing time.
Type:
VT_I8 (lVal)

SMMKVM_TRACKSCOUNT

GUID:
{5A217E85-4359-4834-8B2A-F50A205C0E42}

Description:
Used to get the number of input streams.

Type:
in - VT_I2 (iVal)

SMMKV_DOCTYPE

GUID:
{90B2F4C2-0B5F-4834-92A9-8AC9F086E681}

Description:
Sets the type of output stream. A value of SMMKV_DOCTYPE_ENUM type.

Type:
VT_I4 (lVal)

SMMKV_DOCTYPE_ENUM Enumeration

enum SMMKV_DOCTYPE_ENUM
{
SMMKV_DOCTYPE_MATROSKA = 0,
SMMKV_DOCTYPE_WEBM = 1,
SMMKV_DOCTYPE_SIZE = 2
}

Elements:

SMMKV_DOCTYPE_MATROSKA
Stream type is Matroska.

SMMKV_DOCTYPE_WEBM
Stream type is WEB Matroska.

SMMKV_DOCTYPE_SIZE
Elements number.
SolveigMM Video Editing SDK

SMMKVM_TRACKPROPS

GUID:
{CB2C2D6F-D2F5-49e3-A1B7-F947833DA2CD}

Description:
Used to set the desired tracks properties. Pointer to a TRACK_PROPS structure is used as a parameter.

Type:
VT_BYREF | VT_UI1 (pbVal)

TRACK_PROPS Structure

```c
struct TrackProps {
    int number;
    bool bEnable;
    bool bDefault;
    bool bForced;
    BSTR language;
    BSTR name;
    int track_type;
} TRACK_PROPS;
```

Members:

number
Track's number.

bEnable
Is this track enabled.

bDefault
Is this a default track.

bForced
Is this a forced track.

language
Track's language.

name
Track's name.

track_type
Track's type (read-only value).

SolveigMM MKV Splitter filter

The MKV splitter filter implemented as a DirectShow filter. It can parse Matroska or WEB Matroska files

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
</table>

Solveig Multimedia
Features

The filter's property page is shown on fig. 1. It includes the following control elements:

- Separate pins checkbox - makes one pin for each stream contained in a file.
- A table with the information about file's chapters (if any).

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_MatroskaDemuxer.h header file to be used for tuning the SolveigMM MKV Splitter Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Matroska Splitter DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMKV_H264_ANNEXB</td>
<td>VT_BOOL</td>
<td>Sets/Gets the compatibility with AVC ANNEX B for AVC/H264 video</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SMMKV_SEPARATE_PINS</td>
<td>VT_BOOL</td>
<td>Enables separate pin for each stream.</td>
</tr>
<tr>
<td>SMMV_SETPOSITION</td>
<td>VT_I8</td>
<td>Sets current position (in REFERENCE_TIME).</td>
</tr>
<tr>
<td>SMMKV CHAPTERS</td>
<td>VT_UINT</td>
<td>Gets the information about file chapters.</td>
</tr>
<tr>
<td>SMMKV CHAPTERS</td>
<td>VT_PTR</td>
<td>Gets the information about file chapters.</td>
</tr>
<tr>
<td>SMMKV_DOCTYPE</td>
<td>VT_I4</td>
<td>Gets current file’s type.</td>
</tr>
</tbody>
</table>

**SMMKV_H264_ANNEXB**

**GUID:**

{6F1A0CF5-46C1-4232-815A-17881C912285}

**Description:**

Used to get/set the compatibility with AVC ANNEX B for AVC/H264 video

**Type:**

VT_BOOL (boolVal)

---

**SMMKV_SEPARATE_PINS**

**GUID:**

{56C387DF-68DE-47F4-8135-C8C4EF852B37}

**Description:**

Forces the filter to create one separate pin for each output stream.

**Type:**

VT_BOOL (boolVal)

---

**SMMV_SETPOSITION**

**GUID:**

{04FC1B8B-4642-4022-9269-C9361895CD98}

**Description:**

Sets current position.

**Type:**

VT_I8 (llVal)

---

**SMMKV_DOCTYPE**

**GUID:**
**Description:**
Gets the type of file. Returns a value of `SMMKV_DOCTYPE_ENUM` type.

**Type:**
VT_I4 (IVal)

---

**SMMKV_CHAPTERS**

**GUID:**
{AE39E38E-03DC-4df9-B620-7EF686373239}

**Description:**
Used to read information about file's chapters.

**Type:**
VT_UINT (uintVal)
VT_PTR (pbVal)

**Usage:**
1. Read the number of chapters with VT_UINT.
2. Allocate memory block enough to get the information (Number of chapters * size of ChapterEntry structure).
3. Pass the memory pointer as a pbVal with VT_PTR.
4. Use the pointer as an array of ChapterEntry.

**ChapterEntry Structure**

```c
struct ChapterEntry {
    BSTR name;
    REFERENCE_TIME tStart;
    REFERENCE_TIME tStop;
}
```

**Members:**

- **name**
  Chapter's name.

- **tStart**
  Chapter's start position.

- **tStop**
  Chapter's end position.
**SolveigMM Video Editing SDK**

**SolveigMM MKV Muxer Filter**

The MKV/WEBM muxer filter implemented as a DirectShow filter. It's purpose is to multiplex input video/audio/subtitles into Matroska or WEB Matroska containers.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td></td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_MKV_MUXER</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>Properties - {31741855-F982-40c0-A456-4C8BD7D132A7}, About - {222F030C-2F18-48e8-A8A1-D72FE1C89513}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_MKVMuxer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_MatroskaMuxer.h</td>
</tr>
</tbody>
</table>

**Features**

The filter supports MKV and WEBM as an output formats. The filter's property page is a table shown on fig. 1. It includes the following control elements:

- **DocType** - output stream type (matroska or webm).
- A set of properties for each input stream (including Track's name and language and also enabling/disabling current track).

![Property Page](image)

**Usage**

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM Matroska Muxer Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Matroska Muxer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMKVM_MUXTIME</td>
<td>VT_I8</td>
<td>Reads the current muxing time</td>
</tr>
<tr>
<td>SMMKVM_TRACKSCOUNT</td>
<td>VT_I2</td>
<td>Reads a number of input streams.</td>
</tr>
<tr>
<td>SMMKV_DOCTYPE</td>
<td>VT_I4</td>
<td>Specifies an output stream type.</td>
</tr>
<tr>
<td>SMMKVM_TRACKPROPS</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
</tbody>
</table>

**SMMKVM_MUXTIME**

**GUID:**
{08AE82FB-A9CB-442c-A087-9C59FD2D7E07}

**Description:**
Used to retrieve current muxing time.

**Type:**
VT_I8 (IVal)

**SMMKVM_TRACKSCOUNT**

**GUID:**
{5A217E85-4359-4834-8B2A-F50A205C0E42}

**Description:**
Used to get the number of input streams.

**Type:**
in - VT_I2 (IVal)

**SMMKV_DOCTYPE**

**GUID:**
{90B2F4C2-0B5F-4834-92A9-8AC9F086E681}

**Description:**
Sets the type of output stream. A value of SMMKV_DOCTYPE_ENUM type.

**Type:**
VT_I4 (IVal)

**SMMKV_DOCTYPE_ENUM Enumeration**

```c
enum SMMKV_DOCTYPE_ENUM
{
    SMMKV_DOCTYPE_MATROSKA = 0,
    SMMKV_DOCTYPE_WEBM = 1,
    SMMKV_DOCTYPE_SIZE = 2
}
```

**Elements:**

**SMMKV_DOCTYPE_MATROSKA**
SolveigMM Video Editing SDK

Stream type is Matroska.

**SMMK_DOCTYPE_WEBM**

Stream type is WEB Matroska.

**SMMK_DOCTYPE_SIZE**
Elements number.

---

**SMMKVM_TRACKPROPS**

**GUID:**
{CB2C2D6F-D2F5-49e3-A1B7-F947833DA2CD}

**Description:**
Used to set the desired tracks properties. Pointer to a TRACK_PROPS structure is used as a parameter.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**TRACK_PROPS Structure**

```c
struct TrackProps {
    int number;
    bool bEnable;
    bool bDefault;
    bool bForced;
    BSTR language;
    BSTR name;
    int track_type;
} TRACK_PROPS;
```

**Members:**

- **number**
  Track's number.

- **bEnable**
  Is this track enabled.

- **bDefault**
  Is this a default track.

- **bForced**
  Is this a forced track.

- **language**
  Track's language.

- **name**
  Track's name.

- **track_type**
  Track's type (read-only value).
SolveigMM MKV Splitter Filter

The MKV splitter filter implemented as a DirectShow filter. It can parse Matroska or WEB Matroska files.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, I SpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL; MEDIATYPE_Audio, MEDIASUBTYPE_NULL;</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_MKV_MUXER</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>Properties - {1DE3C022-610A-4528-8B88-0F00BC97E88}, About - {C319D94C-5F60-4a76-9B8A-2836DD99CD81}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_MKVSplitter.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_MatroskaDemuxer.h</td>
</tr>
</tbody>
</table>

Features

The filter's property page is shown on fig. 1. It includes the following control elements:
- Separate pins checkbox - makes one pin for each stream contained in a file.
- A table with the information about file's chapters (if any).

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM MKV Splitter Filter by means of the [IModuleConfig] interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Matroska Splitter DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMKV_H264_ANNEXB</td>
<td>VT_BOOL</td>
<td>Sets/GETs the compatibility with AVC ANNEX B for AVC/H264 video</td>
</tr>
<tr>
<td>SMMKV_SEPARATE_PINS</td>
<td>VT_BOOL</td>
<td>Enables separate pin for each stream.</td>
</tr>
<tr>
<td>SMMV_SETPOSITION</td>
<td>VT_I8</td>
<td>Sets current position(in REFERENCE_TIME).</td>
</tr>
<tr>
<td>SMMKV_CHAPTERS</td>
<td>VT_UINT</td>
<td>Gets the information about file chapters.</td>
</tr>
<tr>
<td>SMMKV_DOCTYPE</td>
<td>VT_I4</td>
<td>Gets current file's type.</td>
</tr>
</tbody>
</table>

**SMMKV_H264_ANNEXB**

**GUID:**
{6F1A0CF5-46C1-4232-815A-17881C912285}

**Description:**
Used to get/set the compatibility with AVC ANNEX B for AVC/H264 video

**Type:**
VT_BOOL (boolVal)

**SMMKV_SEPARATE_PINS**

**GUID:**
{56C387DF-68DE-47F4-8135-C8C4E852B37}

**Description:**
Forces the filter to create one separate pin for each output stream.

**Type:**
VT_BOOL (boolVal)

**SMMV_SETPOSITION**

**GUID:**
{04FC1B8B-4642-4022-9269-C9361895CD98}

**Description:**
Sets current position.

**Type:**
VT_I8 (llVal)

**SMMKV_DOCTYPE**

**GUID:**
{90B2F4C2-0B5F-4834-92A9-8AC9F086E681}

**Description:**
Gets the type of file. Returns a value of `SMMKV_DOCTYPE_ENUM` type.

**Type:**
VT_I4 (lVal)

---

**SMMKV_CHAPTERS**

**GUID:**
{AE39E38E-03DC-4df9-B620-7EF686373239}

**Description:**
Used to read information about file's chapters.

**Type:**
VT_UINT (uintVal)  
VT_PTR (pbVal)

**Usage:**
1. Read the number of chapters with VT_UINT.
2. Allocate memory block enough to get the information( Number of chapters * size of ChapterEntry structure ).
3. Pass the memory pointer as a pbVal with VT_PTR.
4. Use the pointer as an array of ChapterEntry.

**ChapterEntry Structure**

```c
struct ChapterEntry {
    BSTR name;
    REFERENCE_TIME tStart;
    REFERENCE_TIME tStop;
}
```

**Members:**

- **name**
  Chapter's name.

- **tStart**
  Chapter's start position.

- **tStop**
  Chapter's end position.

---

**SolveigMM MP4 Muxer filter**

The MP4/MOV muxer filter implemented as a DirectShow filter. It's purpouse is to multiplex input video/audio/subtitles into MP4 container.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td></td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_MP4_MUXER</td>
</tr>
</tbody>
</table>
Features

The filter supports MP4 and MOV as an output formats.
The filter's property page is a table shown on fig. 1. It includes the following control elements:
- Compatibility - output stream type (ISOM(MP4) or MOV).

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.
To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM Matroska Muxer Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Matroska Muxer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMP4M_MUX_TIME</td>
<td>VT_I8</td>
<td>Reads the current muxing time</td>
</tr>
<tr>
<td>SMMP4_MUX_TYPE</td>
<td>VT_I4</td>
<td>Specifies an output stream type.</td>
</tr>
</tbody>
</table>

SMMP4M_MUX_TIME

GUID:
{SC99C340-3DC0-43c3-A42B-134B73745386}

Description:
Used to retrieve current muxing time.

Type:
VT_I8 (IVal)
SMMP4_MUX_TYPE

GUID: {B5C8AFBC-36C8-4c5c-A464-4D8ED44F8A10}

Description: Sets the type of output stream. A value of Mp4MuxType type.

Type: VT_I4 (IVal)

Mp4MuxType Enumeration

```c
enum Mp4MuxType
{
    MMT_isom = 0,
    MMT_mov = 1,
    MMT_f4v = 2,
    MMT_3gp4 = 3,
    MMT_avc1 = 4,
    MMT_mux_type_cnt = 5
}
```

Elements:

- **MMT_isom**
  Stream type is MP4.

- **MMT_mov**
  Stream type is MOV.

- **MMT_f4v**
  Stream type is F4V.

- **MMT_3gp4**
  Stream type is 3GP4.

- **MMT_avc1**
  Stream type is AVC1.

- **MMT_mux_type_cnt**
  Elements number.

---

SolveigMM MP4 Splitter filter

The MP4 splitter filter implemented as a DirectShow filter. It can parse MP4 or MOV files.

Features

The filter’s property page is shown on fig. 1. It includes the following control elements:
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM MP4 Splitter Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM MP4 Splitter DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMP4_H264_ANNEXB</td>
<td>VT_BOOL</td>
<td>Sets/Gets the compatibility with AVC ANNEX B for AVC/H264 video</td>
</tr>
<tr>
<td>SMMP4_MUX_TYPE</td>
<td>VT_I4</td>
<td>Gets current filte's type.</td>
</tr>
</tbody>
</table>

SMMP4_H264_ANNEXB

GUID: {7F085E94-8E32-4b82-AAA4-6B2EBFF34D43}

Description:
Used to get/set the compatibility with AVC ANNEX B for AVC/H264 video

Type:
VT_BOOL (boolVal)

SMMP4_MUX_TYPE

GUID: {B5C8AFBC-36C8-4c5c-A464-4D8ED44F8A10}

Description:
Gets the type of file. Returns a value of Mp4MuxType type.

Type:
VT_I4 (IVal)
**SolveigMM FLV Muxer filter**

The FLV muxer filter implemented as a DirectShow filter. Its purpose is to multiplex input video/audio/subtitles into FLV container.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td></td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SolveigMMFLVMuxer</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>About - {4A4AAC15-6D05-45df-8683-B91F51FAD4AF}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_FLVMuxer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_FLVMuxer.h</td>
</tr>
</tbody>
</table>

**Features**

The filter supports FLV as an output format.

**Usage**

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_FLVMuxer.h header file to be used for tuning the SolveigMM FLV Muxer Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM FLV Muxer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMFLVM_MUXTIME</td>
<td>VT_I8</td>
<td>Reads the current muxing time</td>
</tr>
</tbody>
</table>

**SMMFLVM_MUXTIME**
GUID:
{53618F8D-6432-4e31-A23F-5D08830EEC05}

Description:
Used to retrieve current muxing time.

Type:
VT_I8 (lVal)

---

## SolveigMM FLV Splitter filter

The FLV splitter filter implemented as a DirectShow filter. It can parse FLV files

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL; MEDIATYPE_Audio, MEDIASUBTYPE_NULL;</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SolveigMMFLVSplitter</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>Properties - {FAAD6812-1FE9-4518-9084-699D44C9AB44}, About - {356969BB-A3F3-4f78-83B2-7338F3A86879}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_FLVDemuxer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_NORMAL</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_FLVDemuxer.h</td>
</tr>
</tbody>
</table>

### Features

The filter's property page is shown on fig. 1. It includes the following control elements:
- ADTS Headers for AAC - Enables Audio Data Transport Stream headers for AAC audio.
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_FLVDemuxer.h header file to be used for tuning the SolveigMM FLV Splitter Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM FLV Splitter DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFLV_H264_ANNEXB</td>
<td>VT_BOOL</td>
<td>Sets/Gets the compatibility with AVC ANNEX B for AVC/H264 video</td>
</tr>
<tr>
<td>SMFLV_AAC_ADTSHEADERS</td>
<td>VT_BOOL</td>
<td>Sets/Gets ADTS Headers output for AAC audio</td>
</tr>
</tbody>
</table>

**SMFLV_H264_ANNEXB**

**GUID:**

{B7A8C7E8-0D0A-4fac-BEEC-96D904BA93CF}

**Description:**

Used to get/set the compatibility with AVC ANNEX B for AVC/H264 video

**Type:**

VT_BOOL (boolVal)

**SMFLV_AAC_ADTSHEADERS**

**GUID:**

{761F0102-05F1-4dd6-97F0-3B4D6B7D7CC7}

**Description:**
SolveigMM ASF Multiplexer filter

The ASF multiplexer filter implemented as a DirectShow renderer filter (file writer). It performs multiplexing input streams to an ASF file and writing output file to a specified path.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages, IFileSinkFilter, IMediaPosition, IMediaSeeking, IWMPProfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL, MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin, IModuleConfig</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMASFMuxFilter</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>CLSID_SMASFMuxSettings, CLSID_SMASFMuxStatistic</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_ASMFxMuxer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_ASMFxMuxer.h</td>
</tr>
</tbody>
</table>

Features

The filter's property page is shown on fig. 1 and includes the following control elements:

- Index ASF checkbox - an option to append index to output file.
- Treat input video as H264 check box. This should be enabled when you deal with AVC/H264 video streams. Also it is possible to change video stream's fourCC.
- PlayReady conformance checkbox - produced stream would be PlayReady conformant.

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of
IPropertyPage interface) to take place, the default values are used otherwise.
To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set necessary properties.
- Run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_ASFMuxer.h header file to be used for tuning the ASF Multiplexer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of ASF Multiplexer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMASFMux_Flags</td>
<td>VT_UI4</td>
<td>Sets/Get multiplexing parameters</td>
</tr>
<tr>
<td>SMASFMux_ResetFlags</td>
<td>VT_UI4</td>
<td>Resets specified parameters.</td>
</tr>
<tr>
<td>SMASFMux_VideoFOURCC</td>
<td>VT_UI4</td>
<td>Specifies custom fourCC for video stream.</td>
</tr>
<tr>
<td>SMASFMux_BFramesMode</td>
<td>VT_I2</td>
<td>Sets the mode to handle B-frames in video.</td>
</tr>
<tr>
<td>SMASFMux_IWMHeaderInfo3Ptr</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMASFMux_ApplyProfile</td>
<td></td>
<td>Applies settings and makes a profile.</td>
</tr>
<tr>
<td>SMASFMux_Language</td>
<td>VT_BSTR</td>
<td>Applies language ID to an input stream.</td>
</tr>
<tr>
<td>SMAT_LogPath</td>
<td>VT_BSTR</td>
<td>Sets the full path to of the log file.</td>
</tr>
<tr>
<td>SMASFMux_Statistics</td>
<td>in - VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMASFMux_CurrentSize</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
</tbody>
</table>

**SMAT_TrimList**

**GUID:**
{51977F10-62C9-46ec-9350-D3CE65E1F34B}

**Description:**
Used to set/retrieve the trimming intervals list. It should be a combination of SMM_ASMuxFlags.

**Type:**
VT_UI4 (ulVal)

**SMM_ASMuxFlags Enumeration**

```c
enum SMM_ASMuxFlags
{
    SMM_ASMuxFlags_WriteIndex = 0x2,
    SMM_ASMuxFlags_InAVC = 0x8,
    SMM_ASMuxFlags_CountSize = 0x10,
    SMM_ASMuxFlags_PlayReady = 0x20
};
```

**Elements:**

**SMM_ASMuxFlags_WriteIndex**
Add index to output file.

**SMM_ASMuxFlags_InAVC**
An input video stream is AVC/H264.

**SMM_ASMuxFlags_CountSize**
Filter doesn't write output file to disk.

**SMM_ASFmuxFlags_PlayReady**
Filter produce PlayReady conformant output file.

---

**SMASFMux_ResetFlags**

**GUID:**
{613F3C29-68BF-4468-BCBE-8F309BB3B45E}

**Description:**
Used to reset specified flags. Should be a combination of SMM_ASFmuxFlags.

**Type:**
VT_UI4 (ulVal)

---

**SMASFMux_VideoFOURCC**

**GUID:**
{AC1A641A-D8B1-48cc-8161-142B3224A697}

**Description:**
Set custom fourCC for video stream.

**Type:**
VT_UI4 (ulVal)

---

**SMASFMux_BFramesMode**

**GUID:**
{39F65152-4B2E-4486-AB2E-2211090ABC79}

**Description:**
Method of B-frames multiplexing. This method is used only when SMM_ASFmuxFlags_InAVC or SMM_ASFmuxFlags_PlayReady flag specified. Should be a member of SMM_BFramesMode.

**Type:**
VT_I2 (iVal)

```c
enum SMM_BFramesMode
{
    SMM_BFrames_USE_DUE = 0,
    SMM_BFrames_Join = 1
}
```

**Elements:**

- **SMM_BFrames_USE_DUE**
B-frames' times are stored in Data Unit Extension.

- **SMM_BFrames_Join**
B-frames are combined with other frames.

---

**SMASFMux_IWMHeaderInfo3Ptr**

**GUID:**
### SolveigMM Video Editing SDK

{AC1A641A-D8B1-48cc-8261-142B3224A695}

<table>
<thead>
<tr>
<th>Function</th>
<th>GUID</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SMASFMux_ApplyProfile</strong></td>
<td>{534D4153-464D-7578-5F41-70706C795072}</td>
<td>Applies settings to a profile. After this call it is possible to get an IWMProfile Interface. (See Windows Media Format SDK Reference for details on interface.)</td>
<td>VT_BYREF</td>
</tr>
<tr>
<td><strong>SMASFMux_Language</strong></td>
<td>{52A82089-2764-49c3-B288-B85C68FC164E}</td>
<td>Applies the language to exact pin. Query the desired pin for the IModuleConfig interface and then use this GUID to set the language ID. For details see the SMM Mux ASF sample application and Windows Media Format SDK Reference.</td>
<td>VT_BSTR (bstrVal)</td>
</tr>
<tr>
<td><strong>SMAT_LogPath</strong></td>
<td>{1F02279E-ECBF-4b22-9D9B-D595DEB6B463}</td>
<td>Set the full path to of the log file.</td>
<td>VT_BSTR (bstrVal)</td>
</tr>
<tr>
<td><strong>SMASFMux_Statistics</strong></td>
<td>{E41A5247-ADEE-4133-82C9-98300459C929}</td>
<td>Reads the processing statistics of video and audio streams. An array of 2 WM_WRITER_STATISTICS structures is used as an input parameter. The first element - a structure describing Video stream statistics and the second is for Audio. See Windows Media Format SDK reference for details on structure.</td>
<td>VT_BYREF</td>
</tr>
<tr>
<td><strong>SMASFMux_CurrentSize</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK

{6B6C20F5-22E5-40ac-B760-8F01C7423E15}

**Description:**
Reads the processing statistics of the file in general. A pointer to WM_WRITER_STATISTICS structure is used as an input parameter. See Windows Media Format SDK reference for details on structure.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

---

**SolveigMM MPA Demuxer filter**

The DirectShow filter for demultiplexing audio from MPEG audio format, and accurate navigation on Variable Bitrate MPEG audio files.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IMediaFilter, IModuleConfig, ISpecifyPropertyPages, IPersistStream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Stream, MEDIASUBTYPE_MPEG1Audio; MEDIATYPE_Audio, MEDIASUBTYPE_NULL;</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_MPA_DEMUXER</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>Properties - {A28CDB74-F1C2-4f08-99FB-8561CCB068DE}, About - {D4E3477C-83BC-4f82-9A14-E3A2840019E3}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_MPADemuxer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_MPADemuxer.h</td>
</tr>
</tbody>
</table>

**Features**

The filter's property page is shown on fig. 1. It includes the following control elements:
- Index radio buttons. The one of the following index behaviors could be chosen:
  - Always load or create index;
  - Load index if exists;
  - Never load index;
- File info section. Shows common parameters of the audio.
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set desired properties for each input streams and the type of output stream.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_MPADemuxer.h header file to be used for tuning the SolveigMM MPA Splitter Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM MPA Splitter DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMPA_INDEX_LOAD</td>
<td>VT_BSTR</td>
<td>Set index file name for loading index from file</td>
</tr>
<tr>
<td>SMMPA_INDEX_SAVE</td>
<td>VT_BSTR</td>
<td>Set index file name for index creation process</td>
</tr>
<tr>
<td>SMMPA_INDEX_STATUS</td>
<td>VT_INT</td>
<td>Start indexation process, stop indexation process, get current progress</td>
</tr>
<tr>
<td>SMMPA_INDEX_MODE</td>
<td>VT_INT</td>
<td>Set index using mode</td>
</tr>
<tr>
<td>SMMPA_INFO</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
</tbody>
</table>

**SMMPA_INDEX_LOAD**

GUID: {0B46233B-52E1-46ec-9E6F-3FFC74F2C6FB}
**Description:**
Set index file name for loading index from file

**Type:**
VT_BSTR (bstrVal)

---

**SMMPA_INDEX_SAVE**

**GUID:**
{436581E9-63CE-45ee-B310-0FFF9493993B}

**Description:**
Set index file name for index creation process

**Type:**
VT_BSTR (bstrVal)

---

**SMMPA_INDEX_STATUS**

**GUID:**
{543A6EC8-F035-489b-BA28-7A03D97453D3}

**Description:**
Start indexation process, stop indexation process, get current progress.

Bits combination is used:

Reading:
0..7 bits - current progress
9 bit - indexation is running

Writing:
8 bit - start indexation process if set to 1
9 bit - stop indexation process if set to 0

**Type:**
VT_INT (intVal)

---

**SMMPA_INDEX_MODE**

**GUID:**
{83262406-A1CA-4379-84FF-7C9CE21B757A}

**Description:**
Set index using mode.
 INDEX_MODE enumeration

typedef enum{
  DO_NOT_USE = 0,
  LOAD_IF_EXIST,
  ALWAYS_LOAD
} INDEX_MODE;

Type:
VT_INT (intVal)

SMMPA_INFO

GUID:
{DCBE345D-7AB8-4f87-AF4B-368108EB3C84}

Description:
Get file information in MPA_FILE_INFO structure

MPA_FILE_INFO structure

typedef struct MPA_FILE_INFO_tag{
  BYTE version;
  BYTE layer;
  DWORD freq;
  DWORD bitrate;
  BYTE channels;
  BYTE vbr;
  DWORD dwReserved[20];
} MPA_FILE_INFO;

Type:
VT_BYREF|VT_UI1 (pbVal)

SolveigMM-Elecard MPEG2 Trimmer filter

SMM_MP2TrimmerFA.ax

The SMM_MP2TrimmerFA.ax is a DirectShow transform filter. It does GOP and frame accurate trimming of MPEG2 video streams and compatible audio streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_MPEG2_VIDEO</td>
</tr>
</tbody>
</table>
Requirements

The filter initially have no audio pins, you should first create the necessary pins.

Features

The filter’s property page is shown on fig. 1 and includes the following control elements:

- Time intervals list and a set of buttons for its editing.
- Edit boxes for specifying start and stop time of each time interval (in DirectShow Reference Time units).
- Edit box for specifying the number of audio pins.
- To get the frame accuracy, tick the "Use index" button and specify the full index file name.

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph and specify the output file name.
- Specify the appropriate audio pins number.

Now the filter can be connected to other filters.

- Set the start and stop positions of the interval and use "Add Time" button to add the interval into the list. Add all necessary intervals like this.
- To get the frame accuracy, tick the "Use index" button and specify the full index file name.
• Build and run the graph.

To use programatically not using property page:
The following section describes the Parameter GUIDs to be used for tuning the SolveigMM-Elecard MPEG2 Trimmer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM-Elecard MPEG2 Trimmer DirectShow Filter.

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM2TFA_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SM2TFA_AudioPinsNum</td>
<td>VT_I4</td>
<td>Sets a number of audio pins to be created</td>
</tr>
<tr>
<td>SM2TFA_Duration</td>
<td>VT_I4</td>
<td>Gets stream duration</td>
</tr>
<tr>
<td>SM2TFA_MPEG2IndexFileName</td>
<td>VT_BSTR</td>
<td>Specifies the index file name.</td>
</tr>
<tr>
<td>SM2TFA_UseMPEG2Index</td>
<td>VT_BOOL</td>
<td>Specifies weather to use the MPEG2 Index file.</td>
</tr>
<tr>
<td>SMAT_3rdParty</td>
<td>VT_BSTR</td>
<td>Specifies the unique GUID to activate the Elecard filters.</td>
</tr>
<tr>
<td>SMAT_TrainAccuracy</td>
<td>VT_I4</td>
<td>Sets/Get the trim accuracy.</td>
</tr>
<tr>
<td>SMAT_SilentPath</td>
<td>VT_BSTR</td>
<td>Sets the silent path.</td>
</tr>
<tr>
<td>SMAT_CodecsSet</td>
<td>VT_BSTR</td>
<td>Sets the profile.</td>
</tr>
</tbody>
</table>

**SM2TFA_TrimList**

GUID: {DC273F41-3A44-425d-82C4-A042D104FE3A}

Description: Used to set/retrieve the trimming intervals list. Pointer to the TrimInfoList structure is used as an input parameter.

Type: VT_BYREF | VT_UI1 (pVal)

**SM2TFA_AudioPinsNum**

GUID: {676545BA-D057-460a-BFD2-BF3F231F8094}

Description: Used to set the desired quantity of audio pins. Default value is 0.

Type: in - VT_I4 (IVal)

**SM2TFA_Duration**

GUID: {9B6F30DC-1E87-41f8-A328-5E8B7D5CAFCB}

Description: Set/Retrieve the type of the output file (member of the SMM_OutputTypes enumeration). If this parameter isn't set the graph wouldn't start.

Type: VT_INT (IntVal)
SM2TFA_MPEG2IndexFileName

**GUID:**
See SMAT_MPEG2IndexFileName

**Description:**
Set/Retrieve the MPEG2 index file name.

**Type:**
VT_BSTR (bstrVal)

SM2TFA_UseMPEG2Index

**GUID:**
{65746723-2112-4ede-9326-F3AD9D658DC4}

**Description:**
Specifies weather to use the MPEG2 Index file.

**Type:**
VT_BOOL (boolVal)

SMAT_3rdParty

**GUID:**
{69F77066-D77C-4c8e-A39B-966E5BD4D17F}

**Description:**
Specifies the unique GUID to activate the Elecard filters. The GUID should be a BSTR value like 
{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}.

**Type:**
VT_BSTR (bstrVal)

SMAT_TrimAccuracy

**GUID:**
{4B07D80A-0FA6-4d69-8438-159CB1BC749F}

**Description:**
Sets/Gets trim accuracy mode. The parameter should be a member of SMM_TrimmAccuracyType.

**Type:**
VT_I4 (lVal)

SMAT_SilentPath

**GUID:**
{2E83B26B-65B3-4609-95B5-E918A96AE151}

**Description:**
Sets the silent path (which is a registry key containing the information about silent filters).

**Type:**
VT_BSTR (bstrVal)
SMAT_CodecsSet

GUID:
{17B25796-46AB-44a6-BBF4-9A3A160E63FB}

Description:
Sets the profile defining which filters should be used.

Type:
VT_BSTR (bstrVal)

SolveigMM ASF Trimmer filter

The ASF trimmer filter implemented as a DirectShow transform filter. It does GOP and Frame accurate trimming of Windows Media video and audio streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMASFTrimmerFilter</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>CLSID_SMASFTrimmerSetPP, CLSID_SMASFTrimmerAboutPP</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_ASF_Trimmer.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_ASFTrimmer.h</td>
</tr>
</tbody>
</table>

Features

The filter's property page is shown on fig. 1 and includes the following control elements:
- Time intervals list and a set of buttons for its editing.
- Edit boxes for specifying start and stop time of each time interval (in DirectShow Reference Time units).
- Checkbox to enable the frame accurate trimming.
Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Set the start and stop positions of the interval and use "Add Time" button to add the interval into the list. Add all necessary intervals like this.
- Set the desired trimming accuracy.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM-Elecard MPEG2 Trimmer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of ASF Trimmer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAT_ASFTrimMode</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SMAT_LogPath</td>
<td>VT_BSTR</td>
<td></td>
</tr>
<tr>
<td>SMAT_TrimAccuracy</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SMAT_Progress_Double</td>
<td>VT_R8</td>
<td></td>
</tr>
</tbody>
</table>

**SMAT_TrimList**

**GUID:**
{C1D0F958-4489-4322-80CD-898089A220C1}

**Description:**
Used to set/retrieve the trimming intervals list. Pointer to the TrimInfoList structure is used as an input parameter.

**Type:**
VT_BYREF | VT_UI1 (pbVal)
SMAT_ASFTrimMode

GUID: {45D7CEAC-89B7-4b90-8370-DBAD5B7DA1BE}

Description: Used to set the desired trimming flags.

Type: in - VT_I4 (IVal)

SMAT_LogPath

GUID: {1F02279E-ECBF-4b22-9D9B-D595DEB6B463}

Description: Set the full path to of the log file.

Type: VT_BSTR (bstrVal)

SMAT_TrimAccuracy

GUID: {4B07D80A-0FA6-4d69-8438-159CB1BC749F}

Description: Used to set the desired trimming accuracy. SMM_TrimAccuracyType is used as a parameter.

Type: in - VT_I4 (IVal)

SMAT_Progress_Double

GUID: {BFBEFA88-33E7-4311-8A5D-34E3B245348B}

Description: Retrieves the current progress value.

Type: VT_R8 (dblVal)

SolveigMM Trimmer filter

The common video-audio trimmer filter implemented as a DirectShow transform filter. It does GOP accurate trimming of AVI-compatible video and audio (including MPEG-1, MPEG-2 audio) streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td></td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_NULL</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_NULL</td>
</tr>
</tbody>
</table>
Features

The filter’s property page is shown on fig. 1 and includes the following control elements:

- Time intervals list and a set of buttons for its editing.
- Edit boxes for specifying start and stop time of each time interval (in DirectShow Reference Time units).
- A set of checkboxes representing filter’s properties.

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Build the graph.
- Set the start and stop positions of the interval and use "Add Time" button to add the interval into the list. Add all necessary intervals like this.
- Run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs to be used for tuning the SolveigMM-Elecard MPEG2 Trimmer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of Trimmer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAT_TrimMode</td>
<td>VT_I4</td>
<td>Obsolete. See SMAT_FLAGS</td>
</tr>
</tbody>
</table>
SolveigMM Video Editing SDK

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_Flags</td>
<td>VT_I4</td>
<td>Sets the trimming flags.</td>
</tr>
<tr>
<td>SMAT_LogPath</td>
<td>VT_BSTR</td>
<td>Specifies the log file name.</td>
</tr>
</tbody>
</table>

**SMAT_TrimList**

**GUID:**
{C1D0F958-4489-4322-80CD-898089A220C1}

**Description:**
Used to set/retrieve the trimming intervals list. Pointer to the TrimInfoList structure is used as an input parameter.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMAT_Flags**

**GUID:**
{AAF40DF7-0698-4424-87DC-3B1E2D60CDD8}

**Description:**
Used to set the desired trimming flags.

**Type:**
in - VT_I4 (IVal)

**SMAT_LogPath**

**GUID:**
{1F02279E-ECBF-4b22-9D9B-D595DEB6B463}

**Description:**
Set the full path to of the log file.

**Type:**
VT_BSTR (bstrVal)

### SolveigMM VP8 Trimmer FA filter

The VP8 trimmer filter is a DirectShow transform filter. It does GOP and frame accurate trimming of VP8 video streams and compatible audio streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td></td>
</tr>
</tbody>
</table>
MEDIATYPE_Video, {30387076-0000-0010-8000-00AA00389B71}  
MEDIATYPE_Audio, MEDIASUBTYPE_VP8 |
| Input Pin Interfaces | IPin, IMemInputPin |
| Output Pin Media Types |
MEDIATYPE_Video, {30387076-0000-0010-8000-00AA00389B71}  
MEDIATYPE_Audio, MEDIASUBTYPE_VP8 |
| Output Pin Interfaces | IPin, IMemInputPin |
Filter CLSID | CLSID_VP8_TRIM_FA
--- | ---
Property Page CLSID | CLSID_VP8_TRIM_FA_PPage, CLSID_VP8_TRIM_FA_About_PPage
Executable | SMM_VP8TrimmerFA.ax
Merit | MERIT_DO_NOT_USE
Filter Category | CLSID_LegacyAmFilterCategory
Declared in | PropID_VP8FATrimmer.h

Requirements

The filter initially have no audio pins, you should first create the necessary pins.

Features

The filter's property page is shown on fig. 1 and includes the following control elements:

- Time intervals list and a set of buttons for its editing.
- Edit boxes for specifying start and stop time of each time interval (in DirectShow Reference Time units).
- Edit box for specifying the number of audio pins.
- To get the frame accuracy, tick the "Frame accuracy mode".

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:

- Add filter to a graph.
- Specify the appropriate audio pins number.
• Set the start and stop positions of the interval and use "Add Time" button to add the interval into the list. Add all necessary intervals like this.
• Build and run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_VP8FATrimmer.h header file to be used for tuning the SolveigMM VP8 Trimmer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM VP8 Trimmer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAT_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SM2TFA_AudioPinsNum</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SMAT_SilenceGen</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAT_SilenceIntervals</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SM2TFA_Duration</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SMAT_3rdParty</td>
<td>VT_BSTR</td>
<td></td>
</tr>
<tr>
<td>SMAT_TrimAccuracy</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SMAT_SilentPath</td>
<td>VT_BSTR</td>
<td></td>
</tr>
<tr>
<td>SMAT_CodecsSet</td>
<td>VT_BSTR</td>
<td></td>
</tr>
</tbody>
</table>

**SMAT_TrimList**

**GUID:**
{C1D0F958-4489-4322-80CD-898089A220C1}

**Description:**
Used to set/retrieve the trimming intervals list. Poiter to the TrimInfoList structure is used as an input parameter.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SM2TFA_AudioPinsNum**

**GUID:**
{676545BA-D057-460a-BFD2-BF3F231F8094}
**SMAT_SilenceGen**

**GUID:**
{653A80BC-901C-4850-8D22-FA0096AD2103}

**Description:**
Sets the IModuleConfig interface of the Silence Generator instance to be used.

**Type:**
in - VT_BYREF | VT_UI1 (pbVal)

---

**SMAT_SilenceIntervals**

**GUID:**
{237C1B77-CF4B-4655-BE9E-C5F413E51F32}

**Description:**
Used to set the array of intervals numbers to be replaced with silence. See the TrimListSilenceNums structure.

**Type:**
VT_BYREF | VT_UI1 (pbVal)

---

**SM2TFA_Duration**

**GUID:**
{9B6F30DC-1E87-41f8-A328-5E8B7D5CAFCB}

**Description:**
Set/Retrieve the type of the output file (member of the SMM_OuputTypes enumeration). If this parameter isn't set the graph wouldn't start.

**Type:**
VT_INT (intVal)

---

**SMAT_TrimAccuracy**

**GUID:**
SolveigMM Video Editing SDK

{4B07D80A-0FA6-4d69-8438-159CB1BC749F}

**Description:**
Sets/Gets trim accuracy mode. The parameter should be a member of `SMM_TrimmAccuracyType`.

**Type:**
VT_I4 (lVal)

---

**SMAT_SilentPath**

**GUID:**
{2E83B26B-65B3-4609-95B5-E918A96AE151}

**Description:**
Sets the silent path (which is a registry key containing the information about silent filters).

**Type:**
VT_BSTR (bstrVal)

---

**SMAT_CodecsSet**

**GUID:**
{17B25796-46AB-44a6-BBF4-9A3A160E63FB}

**Description:**
Sets the profile defining which filters should be used.

**Type:**
VT_BSTR (bstrVal)

---

**SolveigMM AVC Trimmer FA filter**

The MPEG4 AVC trimmer filter is a DirectShow transform filter. It does GOP and frame accurate trimming of AVCHD video streams and compatible audio streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
</table>
| Input Pin Media Types | MEDIATYPE_Video, MEDIASUBTYPE_MPEG2_VIDEO  
MEDIATYPE_Audio, MEDIASUBTYPE_NULL |
| Input Pin Interfaces | IPin, IMemInputPin |
| Output Pin Media Types | MEDIATYPE_Video, MEDIASUBTYPE_MPEG2_VIDEO  
MEDIATYPE_Audio, MEDIASUBTYPE_NULL |
| Output Pin Interfaces | IPin, IMemInputPin |
| Filter CLSID | CLSID_AVC_TRIM_FA |
| Property Page CLSID | CLSID_AVC_TRIM_FA_PPage,  
CLSID_AVC_TRIM_FA_About_PPage |
| Executable | SMM_MPEG4TrimmerFA.ax |
| Merit | MERIT_DO_NOT_USE |
| Filter Category | CLSID_LegacyAmFilterCategory |
| Declared in | PropID_MPEG4FATrimmer.h |
Requirements

The filter initially have no audio pins, you should first create the necessary pins.

Features

The filter’s property page is shown on fig. 1 and includes the following control elements:
- Time intervals list and a set of buttons for its editing.
- Edit boxes for specifying start and stop time of each time interval (in DirectShow Reference Time units).
- Edit box for specifying the number of audio pins.
- To get the frame accuracy, tick the "Frame accuracy mode", "Use index" button and specify the full index file name.

Usage

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Specify the appropriate audio pins number.
- Set the start and stop positions of the interval and use "Add Time" button to add the interval into the list. Add all necessary intervals like this.
- To get the frame accuracy, tick the "Frame accuracy mode" "Use index" button and specify the full index file name.
- Build and run the graph.

To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_MPEG4FATrimmer.h header file to be used for tuning the SolveigMM MPEG4 AVC Trimmer DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM MPEG4 Trimmer DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM2TFA_TrimList</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SM2TFA_AudioPinsNum</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>SM2TFA_Duration</td>
<td>VT_I4</td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SM2TFA_MPEG2IndexFileName</td>
<td>VT_BSTR</td>
<td>Specifies the index file name.</td>
</tr>
<tr>
<td>SM2TFA_UseMPEG2Index</td>
<td>VT_BOOL</td>
<td>Specifies weather to use the MPEG2 Index file.</td>
</tr>
<tr>
<td>SMAT_3rdParty</td>
<td>VT_BSTR</td>
<td>Specifies the unique GUID to activate the Elecard filters.</td>
</tr>
<tr>
<td>SMAT_Tr TrimAccuracy</td>
<td>VT_I4</td>
<td>Sets/Gets the trim accuracy.</td>
</tr>
<tr>
<td>SMAT_SilentPath</td>
<td>VT_BSTR</td>
<td>Sets the silent path.</td>
</tr>
<tr>
<td>SMAT_CodecsSet</td>
<td>VT_BSTR</td>
<td>Sets the profile.</td>
</tr>
<tr>
<td>SM2TFA_TrimList</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GUID:</strong></td>
<td></td>
<td>{DC273F41-3A44-425d-82C4-A042D104FE3A}</td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td></td>
<td>Used to set/retrieve the trimming intervals list. Pointer to the TrimInfoList structure is used as an input parameter.</td>
</tr>
<tr>
<td><strong>Type:</strong></td>
<td></td>
<td>VT_BYREF</td>
</tr>
<tr>
<td>SM2TFA_AudioPinsNum</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GUID:</strong></td>
<td></td>
<td>{676545BA-D057-460a-BFD2-BF3F231F8094}</td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td></td>
<td>Used to set the desired quantity of audio pins. Default value is 0.</td>
</tr>
<tr>
<td><strong>Type:</strong></td>
<td></td>
<td>in - VT_I4 (IVal)</td>
</tr>
<tr>
<td>SM2TFA_Duration</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GUID:</strong></td>
<td></td>
<td>{9B6F30DC-1E87-41f8-A328-5E8B7D5CFCB}</td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td></td>
<td>Set/Retrieve the type of the output file (member of the SMM_OutputTypes enumeration). If this parameter isn't set the graph wouldn't start.</td>
</tr>
<tr>
<td><strong>Type:</strong></td>
<td></td>
<td>VT_INT (intVal)</td>
</tr>
<tr>
<td>SM2TFA_MPEG2IndexFileName</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GUID:</strong></td>
<td></td>
<td>See SMAT_MPEG2IndexFileName</td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td></td>
<td>Set/Retrieve the MPEG2 index file name.</td>
</tr>
<tr>
<td><strong>Type:</strong></td>
<td></td>
<td>VT_BSTR (bstrVal)</td>
</tr>
</tbody>
</table>
SM2TFA_UseMPEG2Index

GUID: {65746723-2112-4ede-9326-F3AD9D658DC4}

Description: Specifies whether to use the MPEG2 Index file.

Type: VT_BOOL (boolVal)

SMAT_3rdParty

GUID: {69F77066-D77C-4c8e-A39B-966E5BD4D17F}

Description: Specifies the unique GUID to activate the Elecard filters. The GUID should be a BSTR value like {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}.

Type: VT_BSTR (bstrVal)

SMAT_TrtrimAccuracy

GUID: {4B07D80A-0FA6-4d69-8438-159CB1BC749F}

Description: Sets/Gets trim accuracy mode. The parameter should be a member of SMM_TrtrimAccuracyType.

Type: VT_I4 (IVal)

SMAT_SilentPath

GUID: {2E83B26B-65B3-4609-95B5-E918A96AE151}

Description: Sets the silent path (which is a registry key containing the information about silent filters).

Type: VT_BSTR (bstrVal)

SMAT_CodecsSet

GUID: {17B25796-46AB-44a6-BBF4-9A3A160E63FB}

Description: Sets the profile defining which filters should be used.

Type: VT_BSTR (bstrVal)
SolveigMM Vorbis Decoder filter

The Vorbis decoder filter implemented as a DirectShow filter. It's purpose is to decode Vorbis-encoded audio streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_Vorbis2</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Audio, MEDIASUBTYPE_IEEE_FLOAT, MEDIATYPE_Audio, MEDIASUBTYPE_PCM</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_Vorbis_Decoder</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>About - CLSID_SMM_Vorbis_Decoder_About_PPage</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_VorbisDecoder.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_NORMAL</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_VorbisDecoder.h</td>
</tr>
</tbody>
</table>

Features

The filter supports Vorbis-encoded audio stream as an input.

Usage

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Run the graph.

SolveigMM VP8 Decoder filter

The VP8 decoder filter implemented as a DirectShow filter. It's purpose is to decode VP8-encoded video streams.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_VP80</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_YV12, MEDIATYPE_Video, MEDIASUBTYPE_I420, MEDIATYPE_Video, MEDIASUBTYPE_UYVY, MEDIATYPE_Video, MEDIASUBTYPE_UYV2, MEDIATYPE_Video, MEDIASUBTYPE_YUY2, MEDIATYPE_Video, MEDIASUBTYPE_YUYV, MEDIATYPE_Video, MEDIASUBTYPE_YVYU</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_VP8_Decoder</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>About - {FB25E68D-BE58-4b42-A322-66801D5DFF95}</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_VP8Decoder.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_NORMAL</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_VP8Decoder.h</td>
</tr>
</tbody>
</table>
**Features**

The filter supports VP8-encoded video stream as an input.

**Usage**

To use the filter manually:
- Add filter to a graph.
- Build the graph.
- Run the graph.

---

**SolveigMM VP8 Encoder filter**

The VP8 Encoder filter is a DirectShow transform filter. It performs video streams compressing to VP8 format.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_YV12</td>
</tr>
<tr>
<td></td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_I420</td>
</tr>
<tr>
<td>Input Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Output Pin Media Types</td>
<td>MEDIATYPE_Video, MEDIASUBTYPE_VP80</td>
</tr>
<tr>
<td>Output Pin Interfaces</td>
<td>IPin, IMemInputPin</td>
</tr>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMM_VP8_ENCODER</td>
</tr>
<tr>
<td>Property Page CLSID</td>
<td>CLSID_SMM_VP8_ENCODER_PPage,</td>
</tr>
<tr>
<td></td>
<td>CLSID_SMM_VP8_ENCODER_About_PPage</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_VP8Encoder.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_VP8Encoder.h</td>
</tr>
</tbody>
</table>

**Features**

The filter supports VP8 as an output video format, which is compatible with WEBM format.

**Usage**

NOTE: All changes must be applied (using either APPLY button in Graph Editor or Apply method of IPropertyPage interface) to take place, the default values are used otherwise.

To use the filter manually:
- Add filter to a graph.
- Build and run the graph.
To use programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_VP8Encoder.h header file to be used for tuning the SolveigMM VP8 Encoder DirectShow Filter by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM VP8 Encoder DirectShow Filter:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMC_Bitrate</td>
<td>VT_I2</td>
<td>Sets the desired bitrate of output video</td>
</tr>
</tbody>
</table>

**SMC_Bitrate**

**GUID:**

{6E4573A1-5E21-4f4b-868A-816DBE66F391}

**Description:**

Used to set/retrieve the bitrate of output video in kB/sec.

**Type:**

VT_I2 (iVal)

---

**SolveigMM Silence Generator Filter**

Silence Generator Filter is implemented as a COM object with ability to add it to DirectShow graph. It can be used to get samples for muted audio with different compression format.

<table>
<thead>
<tr>
<th>Filter Interfaces</th>
<th>IBaseFilter, IModuleConfig, ISpecifyPropertyPages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter CLSID</td>
<td>CLSID_SMSilenceGen</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_SilenceGen.ax</td>
</tr>
<tr>
<td>Merit</td>
<td>MERIT_DO_NOT_USE</td>
</tr>
<tr>
<td>Filter Category</td>
<td>CLSID_LegacyAmFilterCategory</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_SilenceGen.h</td>
</tr>
</tbody>
</table>

**Features**

The SolveigMM Silence Generator supports audio formats: AAC, MPEG Audio, AC3, WMA, PCM, Microsoft ADPCM, Intel ADPCM.

**Usage**

The following section describes the Parameter GUIDs declared in the PropID_SilenceGen.h header file for using SolveigMM Silence Generator by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Silence Generator:
<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMSG_Init</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMSG_SilenceSampleSize</td>
<td>VT_UI4</td>
<td>Get Silence Sample size</td>
</tr>
<tr>
<td>SMSG_SilenceSample</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMSG_SliceSampleDuration</td>
<td>VT_I8</td>
<td>Get Silence Sample Duration</td>
</tr>
</tbody>
</table>

**SMSG_Init**

**GUID:**
{8800F793-07E3-4e7b-8DFE-A055D6ED6B39}

**Description:**
Initialize by SgInitStruct

**SgInitStruct**

```c
struct SgInitStruct {
    BSTR silenceBaseXMLFile;
    BSTR filePathName;
    int streamNum; //1 base
    SMM_OutputTypes fileType;
    BSTR profileString;
    BSTR silentPath;
};
```

**Type:**
VT_BYREF | VT_UI1 (pbVal)

---

**SMSG_SilenceSampleSize**

**GUID:**
{A1683E2B-4508-4dfa-AAE6-7D9B30CC231F}

**Description:**
Get Silence Sample size

**Type:**
VT_UI4 (lVal)
**SMSG_SilenceSample**

**GUID:**
\{41B87714-AD4E-4d9a-B13E-A76C23316065\}

**Description:**
Get Silence Sample data (buffer should be preallocated)

**Type:**
VT_BYREF | VT_UI1 (pbVal)

---

**SMSG_SliceSampleDuration**

**GUID:**
\{3D911100-8A9B-46e8-A690-4EEE1007EA2E\}

**Description:**
Get Silence Sample Duration

**Type:**
VT_I8 (llVal)

---

**SolveigMM Frame Stepper**

The SolveigMM Frame Stepper implemented as COM object. It's purpose is searching next/previous frames/K-frames in video stream from specified time position, getting times of the video frames in specified time range.

**Object Interfaces**

<table>
<thead>
<tr>
<th>Object CLSID</th>
<th>IModuleConfig</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLSID_SMMFrameStepper</td>
<td></td>
</tr>
</tbody>
</table>

**Executable**

SMM_FrameStepper.dll

**Declared in**

PropID_FramesStepper.h

---

**Features**

The SolveigMM Frame Stepper allows searching next/previous frames/K-frames in video stream from specified time position, getting times of the video frames in specified time range.

---

**Usage**

The following section describes the Parameter GUIDs declared in the PropID_FramesStepper.h header file for using the SolveigMM Frame Stepper object by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Frame Stepper object:
### Parameter GUID

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFS_Init</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMFS_GetNextFrameTime</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetPreviousFrameTime</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetPrevOrCrntKFTime</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetLastKFrame</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetLastFrameStop</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetFirstKFrame</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_GetAvgFrameTime</td>
<td>VT_UI8</td>
<td></td>
</tr>
<tr>
<td>SMFS_ReadRange</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMFS_KeyFrameMode</td>
<td>VT_BOOL</td>
<td></td>
</tr>
<tr>
<td>SMFS_Enable</td>
<td>VT_BOOL</td>
<td></td>
</tr>
<tr>
<td>SMFS_UpdateIndex</td>
<td>VT_BSTR</td>
<td></td>
</tr>
</tbody>
</table>

### SMFS_Init

#### GUID:
{769814C6-48AB-4ff5-A615-F84B77CAE484}

#### Description:
Inits object by FMS_InitStruct pointer

#### FMS_InitStruct

```c
typedef struct _FMS_InitStruct
{
    BSTR fileName;
    SMM_OutputTypes fileType;
    BSTR indexFileName;
    DWORD dwAvgFrameTime;
    int nStreamNum;
    BSTR profile;
    BSTR applicationGUID;
    DWORD dwReserved[96];
} FMS_InitStruct;
```
SolveigMM Video Editing SDK

FMS_InitStruct;

Type:
VT_BYREF | VT_UI1 (pbVal)

SMFS_GetNextFrameTime

GUID:
{119A7513-28DC-47db-A675-6964FBCBC8B6}

Description:
Gets next frame time from specified time

Type:
VT_UI8 (llVal)

SMFS_GetPreviousFrameTime

GUID:
{1D2F10CB-1A6D-4832-B740-4C98364D8CF1}

Description:
Gets previous frame time from specified time

Type:
VT_UI8 (llVal)

SMFS_GetPrevOrCrntKFTime

GUID:
{76C303B9-2FEF-44a9-AF37-235BB1008B62}

Description:
Gets frame start time by by specified time

Type:
VT_UI8 (llVal)

SMFS_GetLastKFrame

GUID:
{F9324BE8-01E2-4e93-A661-743FE50FD06D}

Description:
SolveigMM Video Editing SDK

Gets last K frame time

**Type:**
VT_UI8 (llVal)

---

**SMFS_GetLastFrameStop**

**GUID:**
{BA7534AE-DFEE-41b0-92CF-8C4CC85F065A}

**Description:**
Gets last frame stop time

**Type:**
VT_UI8 (llVal)

---

**SMFS_GetFirstKFrame**

**GUID:**
{246EBC3F-B23C-4785-9383-6596BD9D207F}

**Description:**
Gets first K frame time

**Type:**
VT_UI8 (llVal)

---

**SMFS_GetAvgFrameTime**

**GUID:**
{8E891700-793D-4767-87F3-D252E9CD47EC}

**Description:**
Gets avg frame time

**Type:**
VT_UI8 (llVal)

---

**SMFS_ReadRange**

**GUID:**
{4DD67723-B9F9-484d-B9AA-E61B5A9B4C7D}

**Description:**
Gets times array by specified params in FrameTimesRange structure
FrameTimesRange

typedef struct _TimesRange
{
LONGLONG llFrom;
LONGLONG llTo;
ULONG count;
SAFEARRAY* pTimes;
} FrameTimesRange;

Type:
VT_BYREF | VT_UI1 (pbVal)

SMFS_KeyFrameMode

GUID:
{8C3F8D06-0DBE-47f3-BF50-513521017937}

Description:
K-frame mode for SMFS_GetNextFrameTime SMFS_GetPreviousFrameTime if TRUE functions works for K-frames

Type:
VT_BOOL(boolVal)

SMFS_Enable

GUID:
{3E3B0B01-D39D-422d-90DA-9B576DFE1392}

Description:
Restart or Destroy internal graphs

Type:
VT_BOOL(boolVal)

SMFS_UpdateIndex

GUID:
{A27A605E-ED99-4886-B81C-B36D1170A4C8}

Description:
Sets new index file

Type:
VT_BSTR(bstrVal)
SolveigMM Audio Peak Reader

SolveigMM Audio Peak Reader is implemented as a COM object. It is used for getting amplitude peaks data for audio streams for displaying waveform.

<table>
<thead>
<tr>
<th>Object Interfaces</th>
<th>IModuleConfig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object CLSID</td>
<td>CLSID_SMAudioPeakReader</td>
</tr>
<tr>
<td>Executable</td>
<td>SMM_AudioPeakReader.dll</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_AudioPeakReader.h</td>
</tr>
</tbody>
</table>

Features

The SolveigMM Audio Peak Reader allows to get amplitude peaks data for audio streams for displaying waveform.

Usage

The following section describes the Parameter GUIDs declared in the PropID_AudioPeakReader.h header file for using SolveigMM Audio Peak Reader by means of the IModuleConfig interface. The following table provides an overview of the Parameter GUIDs of SolveigMM Audio Peak Reader:

<table>
<thead>
<tr>
<th>Parameter GUID</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMAPR_Init</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAPR_Enable</td>
<td>VT_BOOL</td>
<td></td>
</tr>
<tr>
<td>SMAPR_Run</td>
<td>VT_BOOL</td>
<td></td>
</tr>
<tr>
<td>SMAPR_Range</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAPR_NewPeakCB</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
<tr>
<td>SMAPR_GetPeaksData</td>
<td>VT_BYREF</td>
<td>VT_UI1</td>
</tr>
</tbody>
</table>

SMAPR_Init

GUID: {6014C76E-7229-4fB3-BCB6-BE77F9FE413F}
**Description:**
Inits by APR_InitStruct pointer

**APR_InitStruct**

typedef struct _APR_InitStruct
{
    BSTR fileName;
    SMM_OutputTypes fileType;
    BSTR indexFileName;
    BSTR profileString;
    BSTR applicationGUID;
    int nAudioStreamNum;
    BSTR peaksFileName;
    int buildIgnoreCurPeaksRange;
    DWORD dwReserved[100];
}APR_InitStruct;

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMAPR_Enable**

**GUID:**
{838FB7AF-99E7-4fe4-9BF4-A6CCB960230E}

**Description:**
Builds peaks grab internal graph with decoder

**Type:**
VT_BOOL(boolVal)

**SMAPR_Run**

**GUID:**
{41F8B953-F620-4df8-9EEC-7218D6606FAD}

**Description:**
Runs graph to get peaks

**Type:**
VT_BOOL (boolVal)

**SMAPR_Range**

**GUID:**
{4D002D49-C84D-410b-ADE2-F135B6141059}
**Description:**
Sets bounds for getting peaks by pointer to PEAKS_RANGE

**PEAKS_RANGE**

typedef struct tag_PEAKS_RANGE  
{  
REFERENCE_TIME leftBound;  
REFERENCE_TIME rightBound;  
REFERENCE_TIME priorityStart;  
}PEAKS_RANGE;

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMAPR_NewPeakCB**

**GUID:**
{7AA7770A-BF8A-4db9-B800-960BF8B74C65}

**Description:**
Sets CallBack INewPeakCB that called when new peak found

INewPeakCB : public IUnknown  
{
public:
virtual HRESULT STDMETHODCALLTYPE OnNewPeak( void) = 0;
};

**Type:**
VT_BYREF | VT_UI1 (pbVal)

**SMAPR_GetPeaksData**

**GUID:**
{1B020EE9-EA99-4900-87F3-3E7F54E91828}

**Description:**
SetValue with VT_I4 to set required peaks count  
GetValue will return available peaks data from left bound to right with requested count returns pointer to array of PEAK_MINMAX

typedef struct tag_PEAK_MINMAX  
{
SHORT Min;
}
SolveigMM Thumbnails Generation Library

The SolveigMM Thumbnails Generation Library implemented as dynamic link library in binary (SMM_ThumbNail.dll). It provides methods to grab screenshots from video media files.

Features

The SMM_ThumbNail.dll provides methods to grab screenshots from video media files to JPG and BMP format, with ability to resize image to requested size.

Usage

The SMM_ThumbNail.dll provides the following set of functions:

```c
LPVOID TNCreateNewInstance ( void );
void TNReleaseInstance ( LPVOID pInstance );
void TNSetSilentPathVars ( LPVOID pInstance, const TCHAR* bstProfileString,
       GUID ApplicationGuid, BOOL bAppActivated );
void TNSetMode ( LPVOID pInstance, TN_CONFIG_MODE confMode);
HRESULT TNSetFileName ( LPVOID pInstance, BSTR bstFileName, SMM_OutputTypes
       ft = SMM_File_Type_NO);
HRESULT TNSetIndexFileName ( LPVOID pInstance, const TCHAR* stIndexFileName);
BYTE* TNScreenShot ( LPVOID pInstance, REFERENCE_TIME rtTimePos,
       BITMAPINFOHEADER* pBMPHeader, int * pSize = NULL);
void TNAbort ( LPVOID pInstance );
void TNFreeImageBuffer ( LPVOID pInstance,
       BYTE* pBuffer );
HRESULT TNUseInternalResize ( LPVOID pInstance, BOOL bResize, int nWidth, int
       nHeight );
void TDNDestroyScreenShootGraph( LPVOID pInstance);
void TNDestroyScreen ShootGraph( LPVOID pInstance);
```

- **TNCreateNewInstance** - creates screenshot grabber instance.
- **TNReleaseInstance** - destroys screenshot grabber instance.
- **TNSetSilentPathVars** - init silent path variables (profile string, and application guid to get silent paths for demuxer/decoder components)
- **TNSetMode** - set configuration mode to get various output formats

```c
enum TN_CONFIG_MODE
{
    TN_GETSCREENSHOTS =0, //get Bitmap (bits only)
    TN_GETCOMPRESSEDFRAMES, //get compressed frame data of the source video
    TN_GETJPGSCREENSHOTS, //get JPG format image data to save directly to file
    TN_GETBMPSCREENSHOTS //get BMP format image data to save directly to file
};
```

- **TNSetFileName** - set input media file name
- **TNSetIndexFileName** - set index file name for media file
- **TNSetActiveVStream** - set video stream number
SolveigMM Video Editing SDK

- **TNScreenShot** - get screenshoot at specified time
- **TNAbort** - Abort TNScreenShot function
- **TNFreeImageBuffer** - release buffer that was get by TNScreenShot function
- **TNUseInternalResize** - set image dimensions for output
- **TNDestroyScreenShootGraph** - destroys screenshot graph
- **TNRestoreScreenShootGraph** - restore screenshot graph

---

**SolveigMM Timeline ActiveX Control**

SolveigMM Timeline ActiveX Control's purpose is to display the timeline, show the audio curve and video thumbnails.

<table>
<thead>
<tr>
<th>Object CLSID</th>
<th>CLSID_SMMOCXSlider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executable</td>
<td>SMM_OCXSlider.ocx</td>
</tr>
<tr>
<td>Declared in</td>
<td>PropID_OCXSlider.h</td>
</tr>
</tbody>
</table>

**Features**

The SolveigMM Timeline ActiveX Control provides a user-friendly graphical interface, displaying the timeline, audio wave and video thumbnails. It also allows the user to select video intervals by means of markers.

**Usage**

The following section describes the methods and properties declared in the PropID_OCXSlider.h header file for tuning the SolveigMM Timeline ActiveX Control. The following table provides an overview of the methods and properties of SolveigMM Timeline ActiveX Control:

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>VARIANT_BOOL</td>
<td>Disables/enables the control</td>
</tr>
<tr>
<td>BackColor</td>
<td>OLE_COLOR</td>
<td>Specifies the background color of the control</td>
</tr>
<tr>
<td>ShowZoomButton</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the zoom in/out buttons</td>
</tr>
<tr>
<td>ShowZoomControl</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the zoom control</td>
</tr>
<tr>
<td>ShowEnableThumbnailsButton</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the &quot;Show thumbnails&quot; button</td>
</tr>
<tr>
<td>ShowEnableAudioCurveButton</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the &quot;Show audio curve&quot; button</td>
</tr>
<tr>
<td>ShowThumbNails</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the thumbnails for current file</td>
</tr>
<tr>
<td>ShowAudioCurve</td>
<td>VARIANT_BOOL</td>
<td>Shows/hides the audio curve for current file</td>
</tr>
<tr>
<td>EnableThumbNails</td>
<td>VARIANT_BOOL</td>
<td>Enables/disables thumbnails generation</td>
</tr>
<tr>
<td>EnableAudioCurve</td>
<td>VARIANT_BOOL</td>
<td>Enables/disables audio curve generation</td>
</tr>
<tr>
<td><strong>TimeMode</strong></td>
<td><strong>LONG</strong></td>
<td>Sets the time display mode: 0 - time; 1 - time + frames;</td>
</tr>
<tr>
<td><strong>ShowTimelineButtons</strong></td>
<td><strong>VARIANT_BOOL</strong></td>
<td>Shows/hides additional buttons, which are &quot;Add/Remove Marker&quot;, &quot;Select/Deselect Interval&quot;, &quot;Invert timeline&quot;, &quot;Mute fragment&quot;</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>

**Initialization operations**

| **Init** | Initializes the slider control |
| **MakeActive** | Starts/stops the screen shooting, audio curve construction |

**Trim list operations**

| **GetTrimList** | Retrieves the trim list from the slider control |
| **SetTrimList** | Loads the trim list into the slider control |

**Slider operations**

| **SetPos** | Navigates the slider to the specified time position (in 100nsec Units) |
| **GetPos** | Retrieves the current slider time position (in 100nsec Units) |

**Markers operations**

| **AddMarker** | Inserts new marker to the specified time position (in 100nsec Units) |
| **RemoveMarkerById** | Removes the marker with specified ID |
| **GetMarkersCount** | Retrieves the count of markers |
| **ClearAllMarkers** | Removes all markers |
| **SetMarkerPosition** | Move the marker to the specified time position |
| **GetMarkerPosition** | Retrieves the time position of the marker |
| **GetMarker** | Retrieves the information of specified marker |
| **GetActiveMarkers** | Retrieves the numbers of currently selected markers |
| **RemoveCurrentMarkers** | Removes currently selected markers |
### Intervals operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetIntervalSelect</td>
<td>Marks the interval as &quot;to be saved&quot;</td>
</tr>
<tr>
<td>SetIntervalDeselect</td>
<td>Marks the interval as &quot;to be removed&quot;</td>
</tr>
<tr>
<td>Invert</td>
<td>Inverts the timeline</td>
</tr>
<tr>
<td>GetCurrentIntervalInfo</td>
<td>Retrieves the information about the current interval</td>
</tr>
<tr>
<td>GetCurrentInterval</td>
<td>Retrieves the number of interval selected currently selected by slider</td>
</tr>
<tr>
<td>GetIntervalsCount</td>
<td>Retrieves the intervals count</td>
</tr>
<tr>
<td>GetIntervalInfo</td>
<td>Retrieves the information about the specified interval</td>
</tr>
</tbody>
</table>

### Zoom/scale operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetScale</td>
<td>Gets current scale of timeline</td>
</tr>
<tr>
<td>SetScale</td>
<td>Sets current scale of timeline</td>
</tr>
<tr>
<td>ZoomInOut</td>
<td>Increasing/decreasing the scale to the next/previous predefined step</td>
</tr>
<tr>
<td>SetZoomMinMax</td>
<td>Sets the scale to the maximum/minimum value</td>
</tr>
<tr>
<td>GetMaxScale</td>
<td>Gets maximal internal value</td>
</tr>
</tbody>
</table>

### External menu operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetContextMenu</td>
<td>Sets the menu created externally</td>
</tr>
</tbody>
</table>

### Init

Initializes the slider control

**Syntax:**

```c
void Init (LONG pTimeLineInitStruct);
```

**Parameters:**

- `pTimeLineInitStruct`
  - [in] A pointer to the `TimeLineInitStruct`

### TimeLineInitStruct

```c
struct TimeLineInitStruct
```
Members:

**fileName**
Name of input file.

**fileType**
Type of input file.

**nWidth**
Width of a single thumbnail.

**nHeight**
Height of a single thumbnail.

**profileString**
A string representing current profile.

**indexFileName**
The name of index file for MPEG1-2, AVCHD files.

**applicationGuid**
Application GUID to be used with the silent path and profile.

**nVideoStreamNum**
Number of active video stream.

**nAudioStreamNum**
Number of active audio stream.

**audioPeakFileDir**
Temporary folder to store audio peak files.

**buildCurveIgnoreVisibleArea**
Audio curve building method:
1 to build the curve from the beginning of the file to the end of file,
0 to build the curve from the current position to the end of file

**MakeActive**
Starts/stops the screen shooting, audio curve construction

**Syntax:**
void MakeActive (  
BOOL bActive
);

**Parameters:**
bActive  
[in] Should be set to TRUE to start thumbnail and audio curve generation and FALSE to stop them

---

**SetTrimList**

Loads the trim list into the slider control

**Syntax:**

```c
void SetTrimList (  
VARIANT* pbTrimList
);
```

**Parameters:**
pbTrimList  
[in, out] A pointer to the TrimInfoList, passed as the pbVal of VARIANT

---

**GetTrimList**

Retrieves the trim list from the slider control

**Syntax:**

```c
VARIANT GetTrimList (  
void
);
```

**Return value:**
A pointer to the TrimInfoList, passed as the pbVal of VARIANT

---

**SetPos**

Navigates the slider to the specified time position

**Syntax:**

```c
void SetPos (  
__int64 llPosition
);```
Parameters:
llPosition
[in] Time position in 100 nanosecond Units

GetPos
Retrieves the current slider time position

Syntax:

```c
__int64 GetPos ( 
    void
 );
```

Return value:
Time position in 100 nanosecond Units

AddMarker
Inserts new marker to the specified time position

Syntax:

```c
__int64 AddMarker ( 
    __int64 llPosition
 );
```

Parameters:
llPosition
[in] Time position in 100 nanosecond Units

Return value:
ID of new marker

RemoveMarkerById
Inserts new marker to the specified time position

Syntax:

```c
void RemoveMarkerById ( 
```
__int64 llID
);

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>llID</td>
<td>[in]</td>
<td>Marker ID</td>
</tr>
</tbody>
</table>

### GetMarkersCount

Retrieves the count of markers

**Syntax:**

```c
long GetPos ( void );
```

### ClearAllMarkers

Removes all markers

**Syntax:**

```c
void ClearAllMarkers ( void );
```

### SetMarkerPosition

Move the marker to the specified time position

**Syntax:**

```c
void SetMarkerPosition ( long* lpMarkerNumber, __int64 llPosition );
```

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lpMarkerNumber</td>
<td>[in,out]</td>
<td>The number of marker</td>
</tr>
<tr>
<td>llPosition</td>
<td>[in]</td>
<td>New marker position</td>
</tr>
</tbody>
</table>
GetMarkerPosition

Retrieves the time position of the marker

Syntax:

```c
void GetMarkerPosition (  
long lMarkerNumber,  
__int64* pllPosition  
);
```

Parameters:

- lMarkerNumber  
  [in] The number of marker
- pllPosition  
  [out] Marker position

GetMarker

Retrieves the information of specified marker

Syntax:

```c
void GetMarkerPosition (  
long lMarkerNum,  
VARIANT* pMarkerInfo  
);
```

Parameters:

- lMarkerNum  
  [in] The number of marker
- pMarkerInfo  
  [out] CMarkersInfo pointer which is passed as a pbVal of pMarkerInfo

CMarkersInfo

```c
struct TimeLineInitStruct  
{
    LONGLONG lID;  
    LONGLONG llpos;  
    LONGLONG oldFixedPos;  
    LONGLONG lldata;  
    bool bSelected;  
    bool bValid;  
    int iSelectedNextInterval;  
    DWORD markerFlags;  
};
```
Members:

IID
ID of marker.

llpos
Current position of marker.

oldFixedPos
Previous position of marker.

bSelected
Is marker selected.

iSelectedNextInterval
0 if the interval after marker is marked as "to be removed",
1 if the interval after marker is marked as "to be saved"

markerFlags
The only available flag is MARKERINFO_FLAG_SILENCE which indicates that the interval would be muted.

GetActiveMarkers
Retrieves the information of specified marker

Syntax:

```c
void GetActiveMarkers ( 
    long* lStart, 
    long* lStop 
);
```

Parameters:
    lStart, lStop
    [out] The number of active marker/markers

RemoveCurrentMarkers
Removes currently active markers

Syntax:

```c
void RemoveCurrentMarkers ( 
    void 
);
```
**SetIntervalSelect**

Marks the interval as "to be saved"

**Syntax:**

```c
void SetIntervalSelect(
    long lInterval
);
```

**Parameters:**

- `lInterval` [in] The number of interval

---

**SetIntervalDeselect**

Marks the interval as "to be removed"

**Syntax:**

```c
void SetIntervalDeselect(
    long lInterval
);
```

**Parameters:**

- `lInterval` [in] The number of interval

---

**Invert**

Inverts the intervals selection on the timeline

**Syntax:**

```c
void Invert ( 
    void
);
```

---

**GetCurrentIntervalInfo**

Retrieves the information about the selected interval

**Syntax:**

```c
void GetCurrentIntervalInfo ( 
```
VARIANT* pbIntervalInfo
);

**Parameters:**

pbIntervalInfo
[in, out] CInterval_info pointer which must be passed as a pbVal of pbIntervalInfo

CInterval_info

```
struct CInterval_info
{
   LONGLONG llstart;
   LONGLONG llend;
   bool bselect;
   int N;
   DWORD intervalFlags;
};
```

**Members:**

*llstart*
Start position time of the interval.

*llend*
End position time of the interval.

*bselect*
Is the interval marked as "to be saved".

*N*
The number of interval.

*intervalFlags*
The only available flag is MARKERINFO_FLAG_SILENCE which indicates that the interval would be muted.

---

**GetCurrentInterval**

Retrieves the number of current interval

**Syntax:**

```
long GetCurrentInterval(
void
);
```

**Return value:**
The number of interval

---

**GetIntervalsCount**
SolveigMM Video Editing SDK

Retrieves the total number intervals

**Syntax:**

```c
long GetIntervalsCount(
    void
);
```

**Return value:**
The total number of intervals

---

**GetIntervalInfo**

Retrieves the information about the specified interval

**Syntax:**

```c
void GetIntervalInfo ( 
    long lIntervalNumber 
    VARIANT* pbIntervalInfo 
);
```

**Parameters:**
- `lIntervalNumber` [in] The number of interval
- `pbIntervalInfo` [in, out] CInterval_info structure pointer which must be passed as a pbVal of pbIntervalInfo. See GetCurrentIntervalInfo for structure description.

---

**GetScale**

Gets current scale of the timeline

**Syntax:**

```c
double GetScale(
    void
);
```

**Return value:**
The scale of timeline
**SetScale**

Sets current scale of the timeline

**Syntax:**

```c
void SetScale(
  double dScale
);
```

**Parameters:**

- `dScale`
  - [in] The scale to be set

---

**ZoomInOut**

Increasing/decreasing the scale to the next/previous predefined step

**Syntax:**

```c
void ZoomInOut(
  BOOL bIn
);
```

**Parameters:**

- `bIn`
  - [in] Specifies the type of zooming:
    - TRUE to zoom in,
    - FALSE to zoom out

---

**SetZoomMinMax**

Sets the scale to the maximum/minimum value

**Syntax:**

```c
void SetZoomMinMax(
  BOOL bMin
);
```

**Parameters:**

- `bMin`
  - [in] Specifies the type of zooming:
    - TRUE to set zoom to minimal value,
    - FALSE to set zoom to maximal value
GetMaxScale

Gets maximal internal scale value

Syntax:

double GetMaxScale(
    void
);

Return value:
The maximal internal scale value

SetZoomMinMax

Sets the scale to the maximum/minimum value

Syntax:

void SetZoomMinMax(
    BOOL bMin
);

Parameters:
bMin
[in] Specifies the type of zooming:
TRUE to set zoom to minimal value,
FALSE to set zoom to maximal value

SetContextMenu

Sets the menu created externally

Syntax:

void SetContextMenu(
    long hMenuVal
);

Parameters:
hMenuVal
[in] HMENU of the menu created externally

<table>
<thead>
<tr>
<th>Events</th>
<th>Description</th>
</tr>
</thead>
</table>

Solveig Multimedia
### Slider events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SliderDown</td>
<td>Fired when the user press the left mouse button over the slider</td>
</tr>
<tr>
<td>SliderMoving</td>
<td>Fired when the user is moving the slider</td>
</tr>
<tr>
<td>SliderMoved</td>
<td>Fired when the user releases left mouse button after moving the slider</td>
</tr>
</tbody>
</table>

### Markers events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarkerDown</td>
<td>Fired when the user press the left mouse button over the marker</td>
</tr>
<tr>
<td>MarkerMoving</td>
<td>Fired when the user is moving the marker</td>
</tr>
<tr>
<td>MarkerMoved</td>
<td>Fired when the user releases left mouse button after moving the marker</td>
</tr>
</tbody>
</table>

### Zoom/scale events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleChanged</td>
<td>Fired when the scale is changed</td>
</tr>
</tbody>
</table>

### Menu events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SliderMenuCommand</td>
<td>Fired when the menu command selected</td>
</tr>
</tbody>
</table>

---

**SliderDown**

Fired when the user press the left mouse button over the slider

**Syntax:**

```c
void SliderDown ( void);
```

---

**SliderMoving**

Fired when the user is moving the slider

**Syntax:**

```c
void SliderMoving ( void);
```
SolveigMM Video Editing SDK

**SliderMoved**

Fired when the user releases left mouse button after moving the slider

**Syntax:**

```c
void SliderMoved ( void );
```

**MarkerDown**

Fired when the user press the left mouse button over the marker

**Syntax:**

```c
void MarkerDown ( LONG lMarkerNumber );
```

**Parameters:**

- **lMarkerNumber**
  - [out] The number of marker that is clicked

**MarkerMoving**

Fired when the user is moving the marker

**Syntax:**

```c
void MarkerMoving ( void );
```

**MarkerMoved**

Fired when the user releases left mouse button after moving the marker

**Syntax:**

```c
void MarkerMoved ( void );
```
ScaleChanged

Fired when the scale is changed

Syntax:

```c
void ScaleChanged ( 
    DOUBLE dCurrentScale 
);
```

Parameters:

dCurrentScale
[out] The new scale value

SliderMenuCommand

Fired when the menu command selected

Syntax:

```c
void SliderMenuCommand ( 
    LONG lCmdID 
);
```

Parameters:

lCmdID
[out] Command ID of a menu element that was chosen

SDK Batch file structure

Batch files to be supported by SolveigMM Video Editing SDK are Extensible Markup Language (XML) with 
*.xtl extension and structure based on Direct Show Editing Services XTL format. This section documents 
the XTL elements and attributes.

Note:
All the elements and attributes are case-sensitive.
The values must be quoted(""). 
For an examples see Batch Samples.

This table contains the basic elements of the XTL file:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clip</td>
<td>Specifies a media source file</td>
</tr>
<tr>
<td>group</td>
<td>Defines a group, the top-level object in a timeline. Specifies output file name</td>
</tr>
<tr>
<td>timeline</td>
<td>Defines a timeline. This element is the root node in the XTL file</td>
</tr>
<tr>
<td>track</td>
<td>Defines a track object ( media files )</td>
</tr>
</tbody>
</table>
Here listed all supported attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Possible value</th>
<th>Obligation presence</th>
<th>Belonging to element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Any string excepting symbol (&quot;)</td>
<td>Yes</td>
<td>group</td>
<td>Specifies an output file name with a path</td>
</tr>
<tr>
<td>out_type</td>
<td>[avi, asf, mpa, mpg_ps, mpg_ts, mpg_ves, mpg1_sys, mpg1_ves, wav, avc_ves, matroska, mp4]</td>
<td>No</td>
<td>group</td>
<td>Specifies an output file's type. If not specified - the same as input file.</td>
</tr>
<tr>
<td>mode</td>
<td>[trimming, joining, indexation, multiplexing]</td>
<td>No</td>
<td>group</td>
<td>Specifies the operation type of current task. If not specified - trimming</td>
</tr>
<tr>
<td>video</td>
<td>[0, n]</td>
<td>No</td>
<td>track</td>
<td>Specifies a quantity of video streams</td>
</tr>
<tr>
<td>audio</td>
<td>[0, n]</td>
<td>No</td>
<td>track</td>
<td>Specifies a quantity of audio streams</td>
</tr>
<tr>
<td>accuracy</td>
<td>[gop, frame]</td>
<td>No</td>
<td>track</td>
<td>Specifies a trimming accuracy. If not specified - the best available accuracy type for current format.</td>
</tr>
<tr>
<td>obey_sample_time</td>
<td>[0, 1]</td>
<td>No</td>
<td>track</td>
<td>Specifies to use an alternative synchronization algorithm</td>
</tr>
<tr>
<td>out_type</td>
<td>[avi, asf, mpg_ves, mpg_ps, mpg_ts]</td>
<td>No</td>
<td>track</td>
<td>Specifies an output file format</td>
</tr>
<tr>
<td>src</td>
<td>Any string excepting symbol (&quot;)</td>
<td>Yes</td>
<td>clip</td>
<td>Specifies the input file name with a path</td>
</tr>
<tr>
<td>timeFormat</td>
<td>[100ns_units, time10ms]</td>
<td>No</td>
<td>clip</td>
<td>Specifies the format of &quot;start&quot; and &quot;stop&quot; attributes. If not specified - time10ms</td>
</tr>
<tr>
<td>start</td>
<td>HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units</td>
<td>Yes</td>
<td>clip</td>
<td>Specifies the start time of a fragment to be saved into an output file</td>
</tr>
<tr>
<td>stop</td>
<td>HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units</td>
<td>Yes</td>
<td>clip</td>
<td>Specifies the stop time of a fragment to be saved into an output file</td>
</tr>
</tbody>
</table>
The timeline element defines the timeline. This element is the root node in the XML file.

**Attributes:**
no attributes specified

**Parent/Child information**

<table>
<thead>
<tr>
<th>Parent</th>
<th>Children</th>
</tr>
</thead>
<tbody>
<tr>
<td>None, this is the root element</td>
<td>group</td>
</tr>
</tbody>
</table>

The group element defines a group. The top level object is a timeline.

**Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Possible values</th>
<th>Obligation presence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Any string, except (&quot;) symbol</td>
<td>Yes</td>
<td>Specifies an output filename with a path</td>
</tr>
<tr>
<td>mode</td>
<td>[trimming, joining, indexation, multiplexing]</td>
<td>No</td>
<td>Specifies specifies the operation type of current task. Default value - trimming</td>
</tr>
</tbody>
</table>

**Parent/Child information**

<table>
<thead>
<tr>
<th>Parent</th>
<th>Children</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeline</td>
<td>track</td>
</tr>
</tbody>
</table>

The track element defines an output file.

**Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Possible values</th>
<th>Obligation presence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>video</td>
<td>[0, n]</td>
<td>No</td>
<td>Specifies a quantity of video streams</td>
</tr>
<tr>
<td>audio</td>
<td>[0, n]</td>
<td>No</td>
<td>Specifies a quantity of audio streams</td>
</tr>
<tr>
<td>obey_sample_time</td>
<td>[0, 1]</td>
<td>No</td>
<td>Specifies to use an alternative synchronization algorithm</td>
</tr>
<tr>
<td>out_type</td>
<td>[avi, asf, mpa, mpg_ps, mpg_ts, mpg_ves, mpg1_sys, mpg1_ves, wav, avc_ves, matroska, mp4]</td>
<td>No</td>
<td>Specifies an output file's type. If not specified - the same as input file.</td>
</tr>
<tr>
<td>accuracy</td>
<td>[gop, frame]</td>
<td>No</td>
<td>Specifies a trimming accuracy.</td>
</tr>
</tbody>
</table>
Default value - the best available accuracy type for current format.

Parent/Child information

<table>
<thead>
<tr>
<th>Parent</th>
<th>Children</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>clip</td>
</tr>
</tbody>
</table>

Remarks

For video and audio attributes 0 value means the output file has not to contain a video or audio stream accordingly. By default if neither video nor audio is specified, both of them equate to1. If one of the values presents - default value of missed parameter is 0. By default obey_sample_time equals 0. By default, if out_type is missed, it equals to input file type.

clip Element

The clip element specifies media source files and start/stop positions of fragments to be kept.

Attributes

| Attribute | Possible values                                                                 | Obligation presence | Description                                                                 |
|-----------|--------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------------|---|
| src       | Any string, except (") symbol                                                 | Yes                 | Specifies an input file name with a path                                   |
|           | HH:MM:SS.mS - if timeFormat is time10ms                                       |                     |                                                                             |
|           | 100ns - if timeFormat is 100ns_units                                           |                     |                                                                             |
| start     | HH:MM:SS.mS - if timeFormat is time10ms                                       | Yes                 | Specifies the start time of a fragment to be saved into an output file     |
| stop      | HH:MM:SS.mS - if timeFormat is time10ms                                       | Yes                 | Specifies the stop time of a fragment to be saved into an output file      |
| timeFormat| [100ns_units, time10ms]                                                       | No                  | Specifies the format of "start" and "stop" attributes. If not specified - time10ms |

Parent/Child information

<table>
<thead>
<tr>
<th>Parent</th>
<th>Children</th>
</tr>
</thead>
<tbody>
<tr>
<td>track</td>
<td>None</td>
</tr>
</tbody>
</table>

Batch samples

All XTL samples located in C:\Program Files\Solveig Multimedia\SolveigMM Video Editing SDK\Samples\Media\Batch XTL Samples.

To run the samples you should use the command line console utility SMM_BatchSplit.exe. The full XTL file's path should be passed.

```
SMM_BatchSplit.exe "C:\Program Files\Solveig Multimedia\SolveigMM Video Editing SDK\Samples\Media\Batch XTL Samples\Batch trimming.xtl"
```

In case of Windows 7 the command-line should be run As Administrator, otherwise the resulting files would be stored in VirtualStore folder.

Batch trimming
Remarks
The most important attributes here are:
name - output file,
mode - defines the trimming operation,
video, audio - defines video and audio streams numbers,
frame - defines accuracy type for desired operation,
src - input file,
start, stop, timeFormat - start and stop of each fragment and the format of start and stop values.

As a result the file sample_wmv_0001.wmv should be created

Batch joining

Remarks
The most important attributes here are:
name - output file,
out_type - the type of output file,
mode - defines the joining operation
video, audio - defines video and audio streams numbers
src - input file,
start, stop, timeFormat - start and stop of each fragment and the format of start and stop values.

As a result the file sample_mkv_Joined.mkv should be created

Batch indexation

Remarks
The most important attributes here are:
mode - defines the indexation operation
video, audio - defines video and audio streams numbers
src - input file,
start, stop, timeFormat - start and stop of each fragment and the format of start and stop values.

As a result the file sample_mpeg1_sys.mpg.mlmix should be created
Remarks
The most important attributes here are:
name - output file,
mode - defines the indexation operation
src - input file,
start, stop - for compatibility, should be set to 0.

As a result the file sample_mpeg1_sys.mpg.mlmix should be created

Batch extract video

Batch extract audio

Remarks
This operation's extracts MP3 audio, WMA audio and PCM (to WAV), other audio formats are extracted as is.
The most important attributes here are the same as for trimming, except:
video - should always be set to 0,
out_type - should be set according to the audio format.
start, stop - should be set to 0 and a full length of file accordingly.

As a result the file sample_avi_0001.mp3 should be created

© 2006 – 2013 Solveig Multimedia. All rights reserved.