SolveigMultimedia
software that works

SolveigMM Video Editing SDK

Developer Reference Manual

SDK Version: 2.0 Evaluation
First edition: February 27, 2007
Date modified: May 28, 2009
Note: The documentation of evaluation version is technically restricted

Solveig Multimedia,
Acadimicheskij 10/3, 634055,
Tomsk, Russian Federation
www.solveigmm.com

SolveigMM Video Editing SDK Notices

SDK Version: 2.0

First edition: February 27, 2007

Date modified: May 28, 2009

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 2 492 609 ext. 105; Fax: +7 382 2 492 609

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.

Releases Notes

What's new

New features in SolveigMM Video Editing SDK 2.0

Joining files

The Joining feature can be used to join media files. Currently supported mode: whole file-by-file joining of MPEG audio, AVI, DV-AVI Type 1 and 2, WMA, WMV, ASF, MPEG-1. No reencoding involved, files should have equal parameters.

The component that provides joining ability is SolveigMM Media Joiner Filter.

For more information see SolveigMM Media Joiner Filter and Components description and also Delphi Joiner Sample or Simple Joiner C++ Sample

---

AVC files frame accurate trimming

The current feature is used to implement frame accurate trimming of MPEG-2 Program or Transport stream files that contain AVC/H264 video and any audio streams.

The component that provides the feature is SolveigMM-Elecard MPEG-4 Frame Accurate Trimmer, which uses Elecard AVC Decoder/Encoder filters.

For more information see Components description and AVC Trimming C++ Sample

---

BMP To AVI

SMM BMP to AVI is a set of utilities to operate with the BMP picture files and convert them into AVI file. 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).

The components that provide the feature are SolveigMM BMP Push Source and BMP to AVI DLL

Fore more information see BMP to AVI and BMP To AVI C# Sample

---

ASF Markers

The SDK includes the ability to work with ASF markers - adding, removing, and enumerating markers in exact ASF file.

The component that provides the feature is SolveigMM EditingEngine.
For more information see Editing Engine parameters and SMM ASF Markers sample.

Delphi samples

Included some Delphi and Delphi .NET samples. See SDK Sample Applications

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 files**
  Supported AVI formats: AVI DV type 1,2; OpenDML AVI; AVI files contained audio/video streams encoded by any codecs; AVI files of any size (>2Gbytes and even 4 Gbytes);

- **Multipart GOP-accurate trimming WMV files**
  Supported WMV format: WMV files contained audio/video streams encoded by Microsoft Windows Media codecs.

- **Multipart GOP-accurate trimming ASF files**
  Supported ASF format: ASF files contained audio/video streams encoded by encoded by any third-party codecs.

- **Multipart frame accurate trimming MPEG-2 files**
  Supported MPEG-2 format: Program Stream, Transport Stream.

- **Multipart trimming audio files Supported audio formats: MPEG-1,2 audio; Windows Media Audio.**

- **Re-multiplexing various media formats to Microsoft ASF format without re-encoding.**
  Supported media formats: AVI, Windows Media Format (WMV, WMA, ASF files) MPEG-2 Program Stream, MPEG-2 Transport Stream.

- **Repairing, Indexing damaged or unindexed Windows Media Files (WMV, WMA, ASF files) by means of re-multiplexing.**

- **Extracting audio or video streams from AVI, WMV, ASF files**

- **Processing batch of media files. All media files trimming configurations must be described within a batch file (*.xtl).**

- **Joining media files.**
  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.

Components

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

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

- **SolveigMM Trimmer (SMM_Trimmer.ax)**
  DirectShow filter for AVI, ASF, WMV, WMA, MPEG1, 2 audio (ISO/IEC 11172-3, ISO/IEC 13818-3) GOP accurate files trimming

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

- **SolveigMM-Elecard MPEG2 Frame Accurate Trimmer (SMM_MP2TrimmerFA.ax)**
  DirectShow filter for MPEG-2 Video/Audio files frame accurate trimming.
  The following third party components are involved to MPEG-2 trimming and shipped with this SDK in demo mode
  - Elecard MPEG-2 Video Encoder Std (SD)
  - Elecard MPEG-2 Video Decoder
  - Elecard MPEG Demultiplexer
  - Elecard Multiplexer
  - Elecard Indexator
  - Elecard Index Reader

- **SolveigMM Media Joiner (SMM_MediaJointer.ax)**
  DirectShow filter for joining media 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 BMP Push Source (SMM_BMPPushSrc.Ax)**
  DirectShow filter to make a video stream from chosen BMP files. Currently supported BMP format - 24bit.

- **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

AVITrim
This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features GOP accurate AVI multipart trimming.

AUDTrim
This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MPEG 1,2 audio and Windows Media audio trimming.

ASFMux
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.

BatchSplit
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.

MPEG2Trim
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 filers

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

AVCTrim
This sample application is an analogue of MPEG2Trim sample but 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 Hd
- Elecard AVC Video Decoder
- Elecard MPEG Demultiplexer
- Elecard Multiplexer
- Elecard Indexator
- Elecard Index Reader

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

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

ASFMarkers
This sample application demonstrates using SolveigMM Video Editing Engine. It illustrates ASF markers workflow.

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

DelphiTrimmer
This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of files with supported formats

DelphiNetTrimmer
This is a Delphi .NET analogue of DelphiTrimmer sample. It is intended to show the usage of Editing Engine in .NET environment with Delphi programming language.
The sample depends on the following components:

- DShowAddonTypeLib.dll
- Interop.AVITRIMMERCOMLib.dll
- Interop.QuartzTypeLib.dll

This sample demonstrates using SolveigMM Video Editing Engine via VB.60 interfaces. It illustrates GOP accurate media files multipart trimming.

VB.NET Trim Sample

This sample demonstrates using SolveigMM Video Editing Engine within VB.NET. It illustrates GOP accurate media files multipart trimming.

System requirements

- CPU (Intel® Pentium II, Celeron, AMD® Athlon, Opteron etc.)
- 128 MB RAM.
- Any VGA card.
- Windows® 98/Me/2000/XP.
- Microsoft® DirectX® Media SDK or Microsoft® DirectX® 8.1 SDK.
- Microsoft® Windows Media Format 9 Series Runtime® (to be shipped with Windows Media Player 9 ® and higher)

SolveigMM Video Editing SDK User Guide

Installing SDK

To install the SolveigMM Video Editing SDK:

1. Run the SDK setup. To run, double click the executable file from the SDK setup package.
2. The *SolveigMM Video Editing SDK* window will appear. Read the recommendations and warnings. Click **Next**.
3. The Release Notes will appear. Click **Next**.
4. The license agreement will appear. Read the agreement and if you accept the terms within, select the *"Yes I agree with the terms of this license agreement"* check box. Click **Next**.
5. Select the destination folder in which you want to install the SDK. Click **Next**.
6. Select the program group in which you want the SDK to be located. Click **Next**.
7. 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->Uninstall SDK*.

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

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
Doc - includes all SDK-related documentation
Include- includes headers to comprise interfaces and GUIDS headers

- **Include\ElecardHeaders\Inc**
- **Include\ElecardHeaders\Classes\ -** includes Elecards SDK headers to be used by the samples for creating DirectShow filter graph only

Lib - includes Elecard SDK debug\release libraries to be used by the samples for creating DirectShow filter graph only
Samples – includes VC++ 7.0, VB.NET, VB.60, Delphi, Delphi .NET sample applications

SDK Sample Applications

AVITrim
This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features GOP accurate AVI multi-parts trimming.

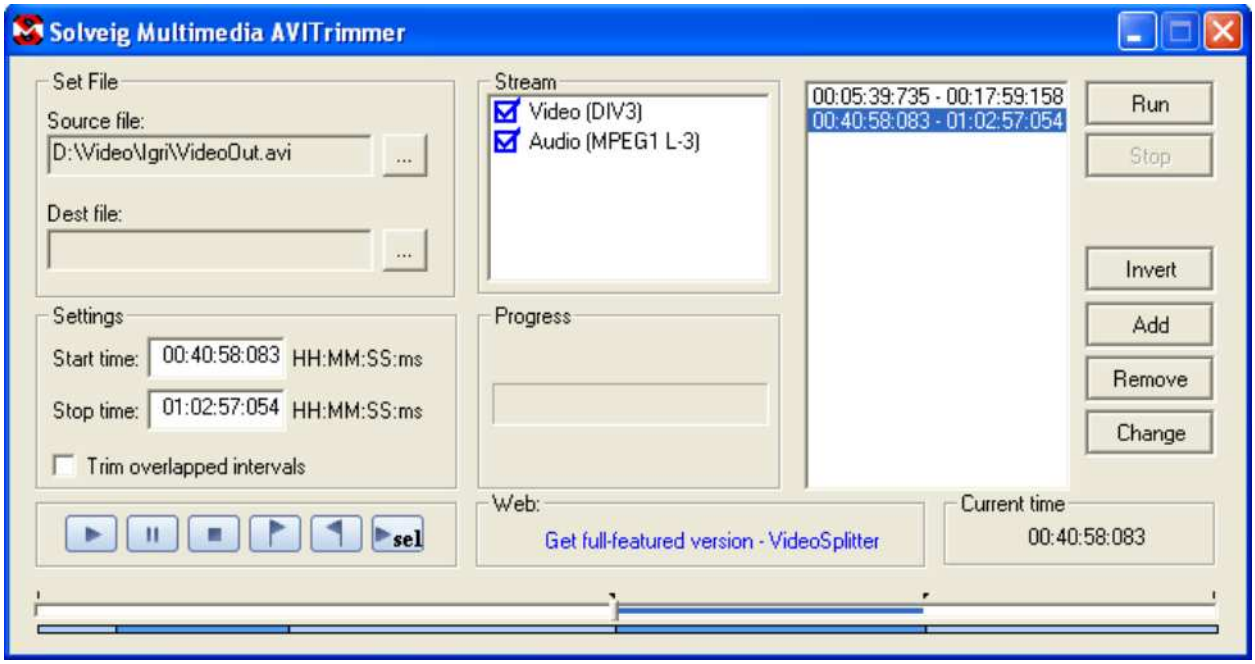


Figure 1 – AVITrim sample application

AUDTrim
This sample demonstrates using SolveigMM Video Editing Engine. It illustrates one of the engine features MPEG 1,2 audio and Windows Media Audio files trimming.

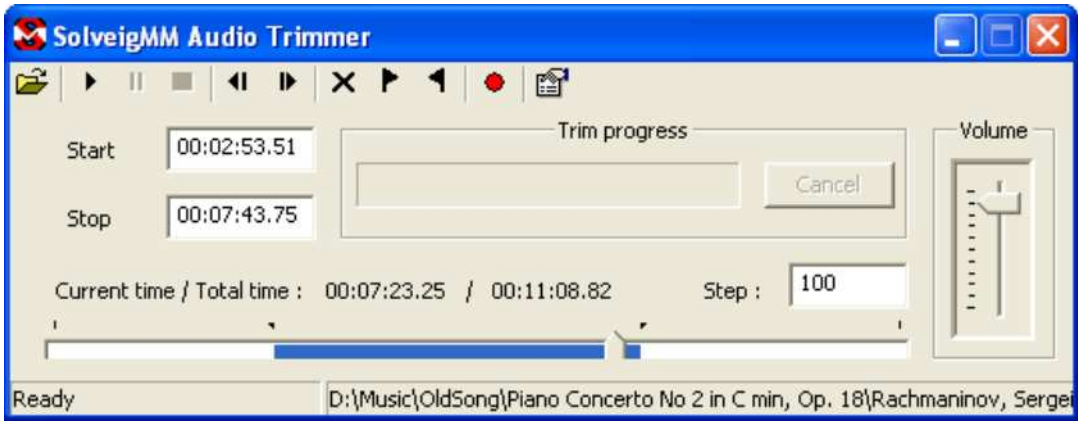


Figure 2 – AUDTrim sample application

ASFMux
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.

Figure 3 – ASFMux sample application

BatchSplit

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.

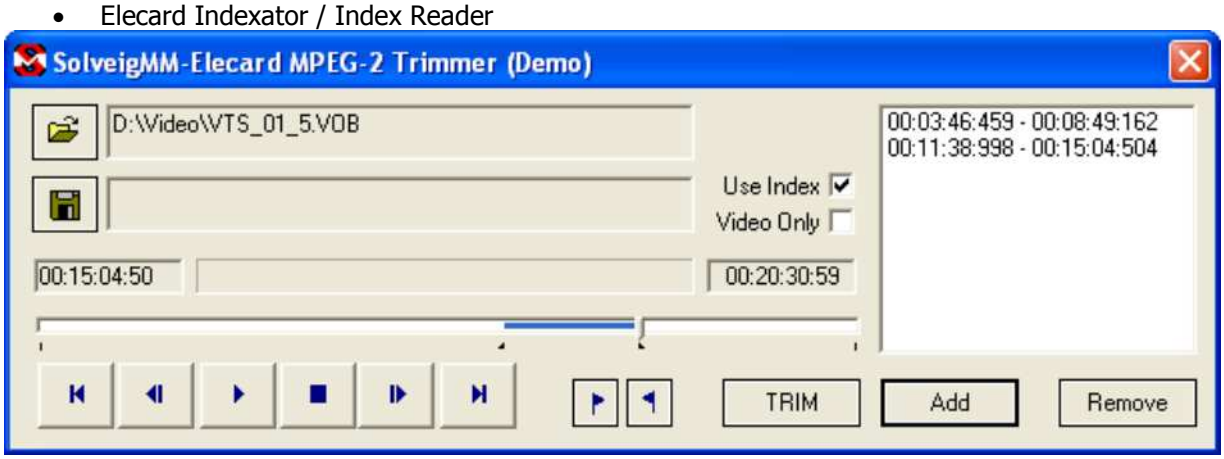
ASFMarkers

This sample application demonstrates using SolveigMM Video Editing Engine. It illustrates ASF markers workflow.

MPEG2Trim

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 to be registered

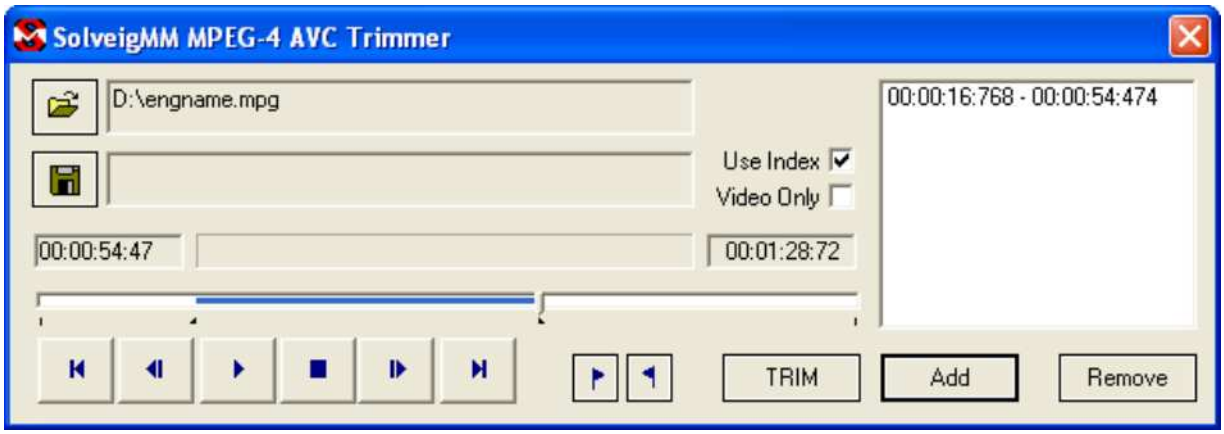
- Elecard MPEG-2 Video Encoder Std
- Elecard MPEG-2 Video Decoder
- Elecard MPEG Demultiplexer
- Elecard Multiplexer
- Elecard Indexator / Index Reader



AVCTrim

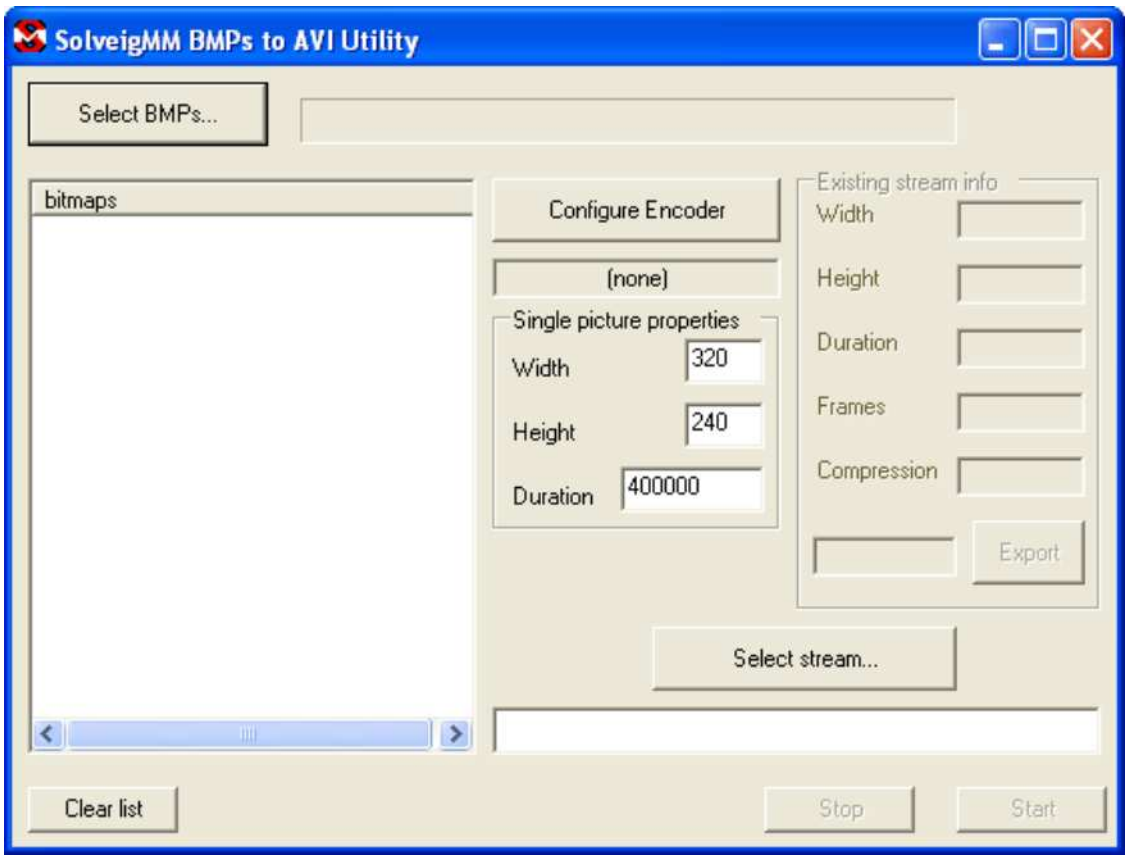
This sample application is an analogue of MPEG2Trim sample but demonstrates usage of SolveigMM MPEG-4 Frame Accurate Trimmer DirectShow Filter. It illustrates frame accurate trimming of AVC/H264 streams wrapped in the Program or Transport streams. It requires the following DirectShow filters:

- Elecard AVC Video Encoder Hd
- Elecard AVC Video Decoder
- Elecard MPEG Demultiplexer
- Elecard Multiplexer
- Elecard Indexator
- Elecard Index Reader



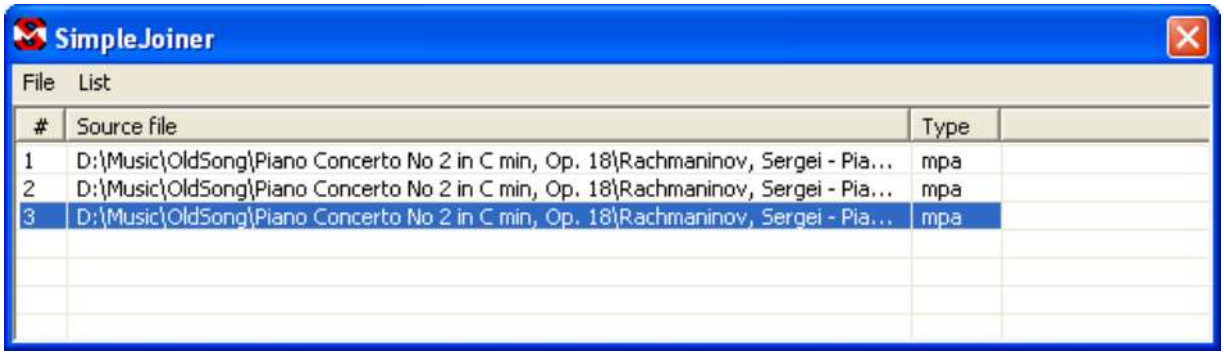
SMM BMP to AVI Utility

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



SimpleJoiner

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

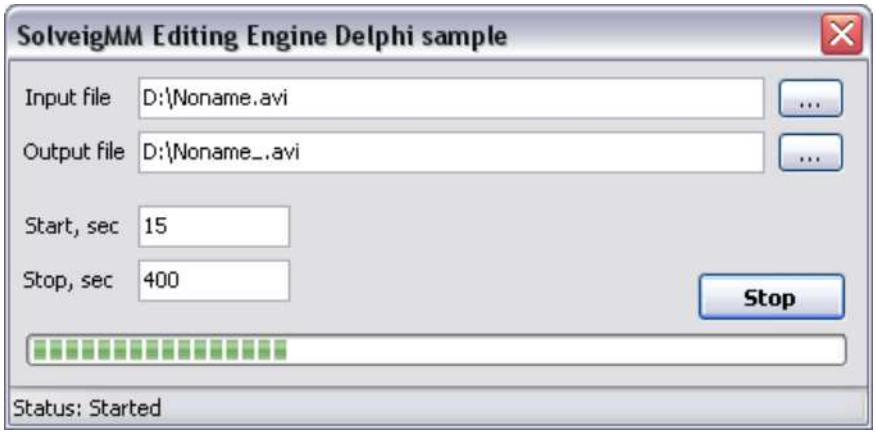


DelphiJoiner

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

DelphiTrimmer

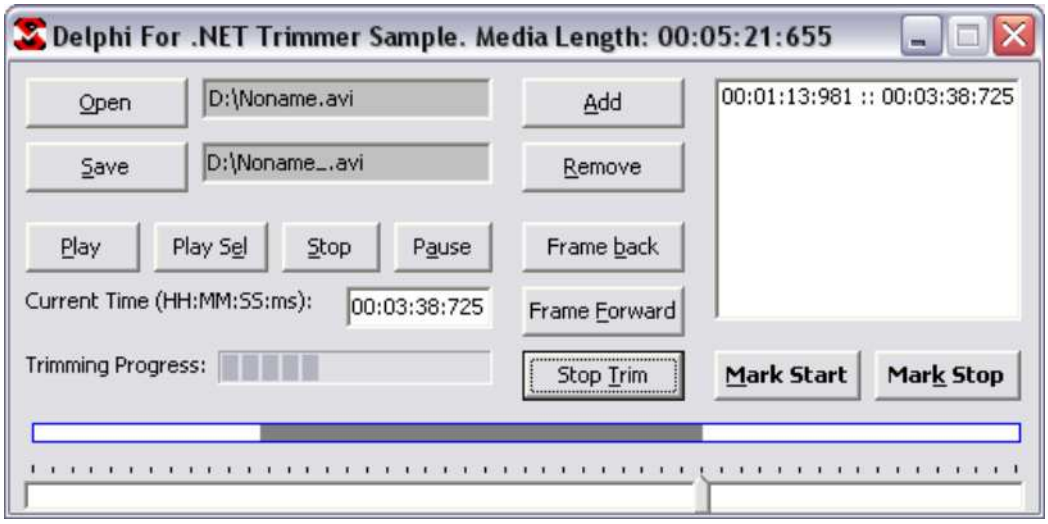
This is a Delphi Sample demonstrating the usage and tuning of SMM_EditingEngine to implement trimming of files with supported formats



DelphiNetTrimmer

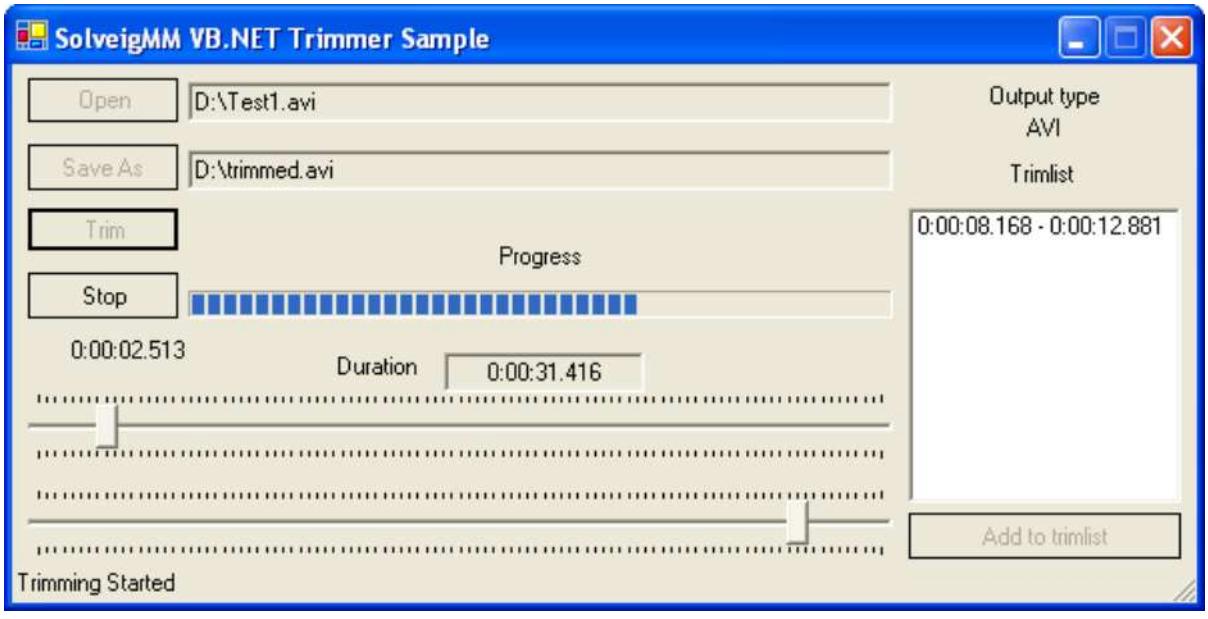
This is a Delphi .NET analogue of DelphiTrimmer sample. It is intended to show the usage of Editing Engine in .NET environment with Delphi programming language.
The sample depends on the following complnents:

- DShowAddonTypeLib.dll
- Interop.AVITRIMMERCOMLib.dll
- Interop.QuartzTypeLib.dll



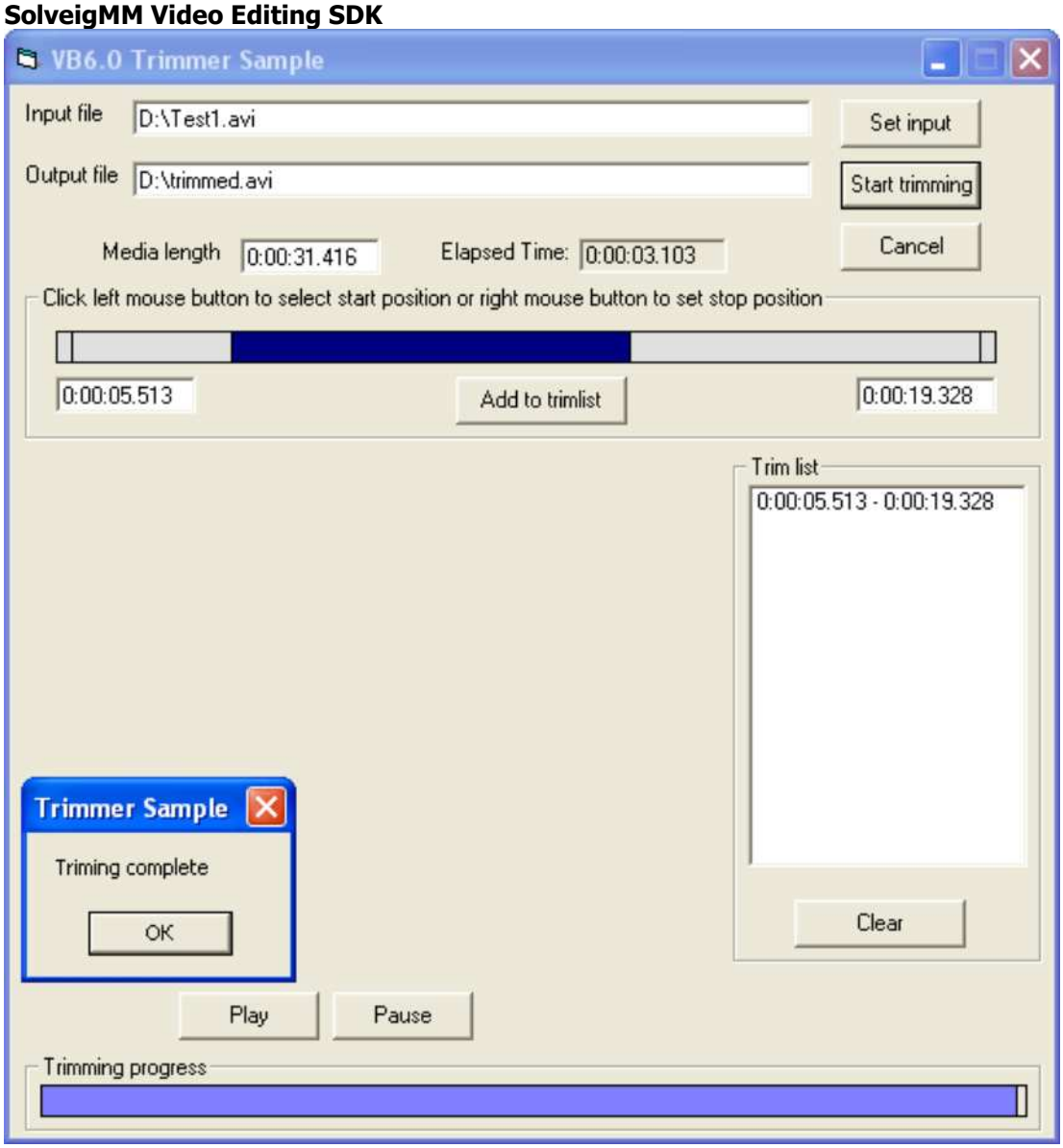
VB.NET Trim Sample

This sample demonstrates using SolveigMM Video Editing Engine within VB.NET. It illustrates GOP accurate media files multipart trimming.



VB6.0 Trim Sample

This sample demonstrates using SolveigMM Video Editing Engine via VB.60 interfaces. It illustrates GOP accurate media files multipart trimming.


VB6.0 Trimmer Sample
Input file
D:\Test1.avi
Set input
Output file
D:\trimmed.avi
Start trimming
Media length
0:00:31.416
Elapsed Time:
0:00:03.103
Cancel
Click left mouse button to select start position or right mouse button to set stop position
0:00:05.513
Add to trimlist
0:00:19.328
Trim list
0:00:05.513 - 0:00:19.328
Trimmer Sample
Triming complete
OK
Clear
Play
Pause
Trimming progress

SolveigMM Video Editing SDK Reference Manual

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

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

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

IUnknown methods	**Description**
QueryInterface	Retrieves pointers to supported interfaces
AddRef	Increments reference count
Release	Decrements reference count
IDispatch methods	**Description**
GetTypeInfoCount	Retrieves the number of type information interfaces that an object provides (either 0 or 1)
GetTypeInfo	Gets the type information for an object
GetIDsOfNames	Maps a single member and an optional set of argument names to a corresponding set of integer DISPIDs
Invoke	Provides access to properties and methods exposed by an object
ITrimmerObjControl methods	**Description**
Start	Starts the engine trimming process
Stop	Stops the engine trimming process
put_InputFN	Obsolete. Substituted IModuleConfig interface with SMAT_InputFile guid
get_InputFN	Obsolete. Substituted IModuleConfig interface with SMAT_InputFile guid
put_OutputFN	Obsolete. Substituted IModuleConfig interface with SMAT_OutputFile guid
get_OutputFN	Obsolete. Substituted IModuleConfig interface with SMAT_OutputFile guid
get_Status	Obsolete. Substituted IModuleConfig interface with SMAT_Progress guid
get_StreamLength	Obsolete.
put_StartPosition	Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid
get_StartPosition	Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid
put_StopPosition	Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid
get_StopPosition	Obsolete. Substituted IModuleConfig interface with SMAT_TrimList guid
SetCallback	Obsolete. Substituted IModuleConfig interface with SMAT_Callback guid

ITrimmerObjControl::Start

ITrimmerObjControl interface

The **Start** method starts the engine with the parameters set

Syntax

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

Return Value

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

ITrimmerObjControl::Stop
ITrimmerObjControl interface

The **Stop** method stops the engine trimming process
Syntax

```
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

IUnknown methods	**Description**
QueryInterface	Retrieves pointers to supported interfaces
AddRef	Increments reference count
Release	Decrements reference count
ITrimmerObjControlCB methods	**Description**
OnStart	Callback method to indicate the engine started trimming process
OnStop	Callback method to indicate the engine stopped trimming process
OnError	Callback method to indicate that some error occurred
OnPause	Not implemented

ITrimmerObjControlCB::OnStart
ITrimmerObjControlCB interface

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

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

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

ITrimmerObjControlCB::OnStop
ITrimmerObjControlCB interface

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

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

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

---

ITrimmerObjControlCB::OnError
ITrimmerObjControlCB interface

The **OnError** callback method is called indicating some error occurred
Syntax

```
HRESULT OnStop  (
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

Editing Engine enumerations

SMM_OutputTypes
Indicates the type of the file.

```
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,
    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).

```
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.

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

```
class SMM_TRACK_INFO {
public:
    int                   nVideoStream;
    int                   nAudioStream;
    WCHAR                 szMPEG2IdxFileName[1024];
    SMM_CLIP_INFO         *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_TRACK_INFO 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.

llASValue
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)

```
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, which means you are going to append the whole file to the output. This is currently the only mode for joining, otherwise the error will occur!

TrimInfoList(Structure)
Description:
Used to set trimming parameters to SolveigMM Video Editing Engine through SMAT_TrimList parameter.

```
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 informatoin on filters used.

```
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.

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

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. Table 1 provides an overview of the Parameter GUIDs for the SolveigMM Video Editing Engine.

Table 1. SolveigMM Video Editing Engine Parameters GUIDs

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

Parameter GUID	Value Type	Available Values	Default	Description
				MPEG2 index filename.
SMAT_SilentPath	VT_BSTR		-	Sets the registry path that stores paths of filters to use in silent mode
SMAT_MPEG2CodecsSet	VT_BSTR		See below	Sets the set of codecs to use for MPEG2 files editing
SMAT_GetFileType	*See below	[SMM_OutputTypes]		Returns the type of the file
SMAT_ValidateFiles	VT_I4	HRESULT values		Validate the parameters of chosen files.
SMAT_ASFMarker	VT_BYREF	-	-	Sets or retrieves ASF marker with specified name and position
SMAT_ASFMarkersCount	VT_I4	Read only	-	Retrieves a count of markers contained in an input ASF
SMAT_ASFMarkerRemove	VT_BSTR, VT_I4	-	-	Removes a marker from an input ASF

SMAT_InputFile

GUID:
Available in the full SDK version

Description:
Specifies an input media file name (AVI, WMV, ASF, MP3, MPA, WMA). For details please see AVITrim, AudTrim, BatchSplit sample applications.

Type:
VT_BSTR

SMAT_OutputFile

GUID:
Available in the full SDK version

Description:
Specifies an output media file name the Engine saves processed data to. For details please see AVITrim, AudTrim, BatchSplit sample applications.

Type:
VT_BSTR

SMAT_Flags

GUID:
Available in the full SDK version

Description:
Set trimming configuration flags to the Video Editing Engine. For details please see AudTrim sample application.

Type:
VT_UI4

Note:
Value 0 means reset all flags.

SMAT_ResetFlags

GUID:
Available in the full SDK version

Description:
Resets specified flags that were set before by means of SMAT_Flags parameter. For details please see AudTrim sample application.

Type:
VT_UI4

SMAT_TrimList

GUID:
Available in the full SDK version

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 AVITrim, AudTrim sample applications.

Type:
VT_BYREF | VT_UI1

SMAT_Progress

GUID:
Available in the full SDK version

Description:
Retrieves a progress of trimming process in percentage. For details please see AVITrim, AudTrim sample applications.

Type:
VT_I4

Available Values:
0 – 100

SMAT_AudioStreamNumber

GUID:
Available in the full SDK version

Description:
Sets a number of an audio stream to be trimmed.

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:
Available in the full SDK version

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:
Available in the full SDK version

Description:
Sets a callback COM interface – ITrimmerObjControlCB. For details please see example code and BatchSplit, AVITrim 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:
Available in a FULL version of SDK

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;
hr         = m_pIModuleConfig->GetValue( &SMAT_GetVersions2, &var );
```

Type:
VT_BYREF | VT_UI1

SMAT_CurrentSize

GUID:
Available in the full SDK version

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

Type:
VT_UI8

SMAT_BatchFileName

GUID:
Available in the full SDK version

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:
Available in the full SDK version

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:
Available in the full SDK version

Description:
Specifies output file type as SMM_OutputTypes parameter

Type:
VT_UI4

SMAT_BatchTasksNum

GUID:
Available in the full SDK version

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

Type:
VT_UI4

SMAT_TaskType

GUID:
Available in a FULL version of SDK

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

Type:
VT_UI4

Available values:
see Editing Engine enumerations and structures.

SMAT_TrackInfo

GUID:
Available in a FULL version of SDK

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).

Type:
VT_BYREF | VT_UI1

SMAT_MPEG2IndexFileName

GUID:
Available in a FULL version of SDK

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

Type:
VT_BSTR

SMAT_SilentPath

GUID:
Available in a FULL version of SDK

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

Type:
VT_BSTR

SMAT_MPEG2CodecsSet

GUID:
Available in a FULL version of SDK

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).

Type:
VT_BSTR

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_GetFileType

GUID:
Available in a FULL version of SDK

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.

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

Example Code:

```
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;
```

SMAT_ValidateFiles

GUID:
Available in a FULL version of SDK

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

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;
hr = m_pIModuleConfig->SetValue( &SMAT_ValidateFiles, &var );
if( SUCCEEDED(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 SimpleJoiner 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:

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

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 abcense 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).

SMAT_ASFMarker

GUID:
Available in FULL version of SDK

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:
Available in FULL version of SDK

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

Type:
VT_I4

SMAT_ASFMarkerRemove

GUID:
Available in FULL version of SDK

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

The SMM_MP2TrimmerFA.ax is a DirectShow transform filter. It does GOP and frame accurate trimming of MPEG2-files.

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.

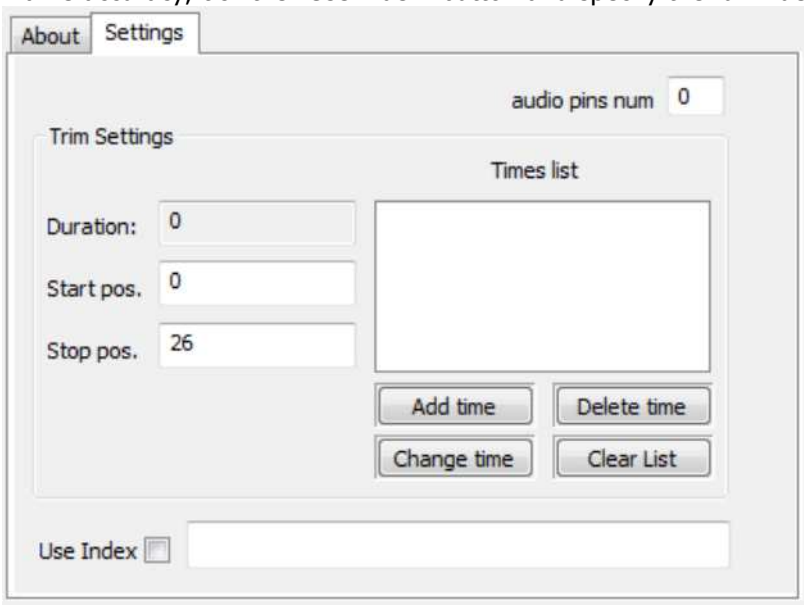


Fig 1.

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 programmatically not using property page:

The following section describes the Parameter GUIDs declared in the PropID_MP2FATrimmer.h header file 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.

Parameter GUID	Value type	Description
SM2TFA_TrimList	VT_BYREF |VT_UI1	Sets trimming parameters
SM2TFA_AudioPinsNum	VT_I4	Sets a number of audio pins to be created
SM2TFA_Duration	VT_I4	Gets stream duration
SM2TFA_MPEG2IndexFileName	VT_BSTR	Specifies the index file name.
SM2TFA_UseMPEG2Index	VT_BOOL	Specifies weather to use the MPEG2 Index file.
SMAT_3rdParty	VT_BSTR	Specifies the unique GUID to activate the Elecard filters.
SMMJ_ExternalMessageWindow	-	Set the external window to receive the graph messages.

SM2TFA_TrimList

GUID:
Available in a FULL version of SDK

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)

SM2TFA_AudioPinsNum

GUID:
Available in a FULL version of SDK

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

Type:
in - VT_I4 (lVal)

SM2TFA_Duration

GUID:
Available in a FULL version of SDK

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:
Available in a FULL version of SDK

Description:
Set/Retrieve the MPEG2 index file name.

Type:

VT_BSTR (bstrVal)

SM2TFA_UseMPEG2Index

GUID:
Available in a FULL version of SDK

Description:
Specifies weather to use the MPEG2 Index file.

Type:
VT_BOOL (boolVal)

SMAT_3rdParty

GUID:
Available in a FULL version of SDK

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

Type:
VT_BSTR (bstrVal)

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 reading 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 writing the BMP into AVI file and also getting 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 these 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, and 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 then 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

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.

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. 2) includes the following control elements:

- Table containing the list BMP files,
- Add File button – opens the "Open File" dialog to choose the desired files (alternatively drag-n-drop can be used).
- DeleteChosen – 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 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 it would be skipped.

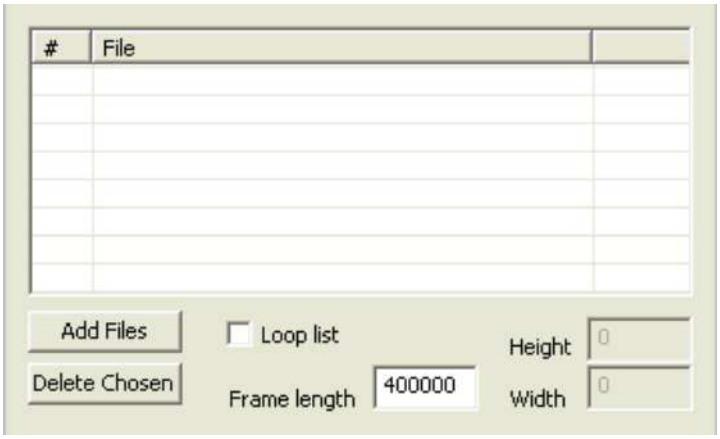


Fig 2

Usage.

NOTE: All changes must be applied (using APPLY button in Graph Editor or Apply method of IPropertyPage 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:

Parameter GUID	**Value type**	**Description**
SMMBPS_StrmProps	VT_BYREF | VT_UI1	Set/retrieve the stream parameters
SMMBPS_FilesTabl	VT_BYREF | VT_UI1	Set/retrieve the list of files
SMMBPS_LoopList	VT_BOOL	
SMMBPS_InputBMP	-	BMP to be appended
SMMBPS_IsQueue	VT_BOOL	Checks if the filter's internal samples queue is empty
SMMBPS_Terminate	-	Terminates the stream creation and stops graph

SMMBPS_StrmProps

GUID:
Available in a FULL version of SDK

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:
Available in a FULL version of SDK

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

Type:
VT_BYREF | VT_UI1 (pbVal)

SMMBPS_LoopList

GUID:
Available in a FULL version of SDK

Description:
Duplicates the property page control.

Type:
VT_BOOL (boolVal)

SMMBPS_InputBMP

GUID:
Available in a FULL version of SDK

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:
Available in a FULL version of SDK

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

Type:
VT_BOOL (boolVal)

GUID:
Available in a FULL version of SDK

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

Type:
not applied

Here is the definition of StreamParameters structure

```
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 using 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.
- 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.

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

Requirements.

The filter currently supports the following formats: AVI, Windows Media files (ASF, WMV, WMA), MPEG1 Audio (Layer 1-3). 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. 3) 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).

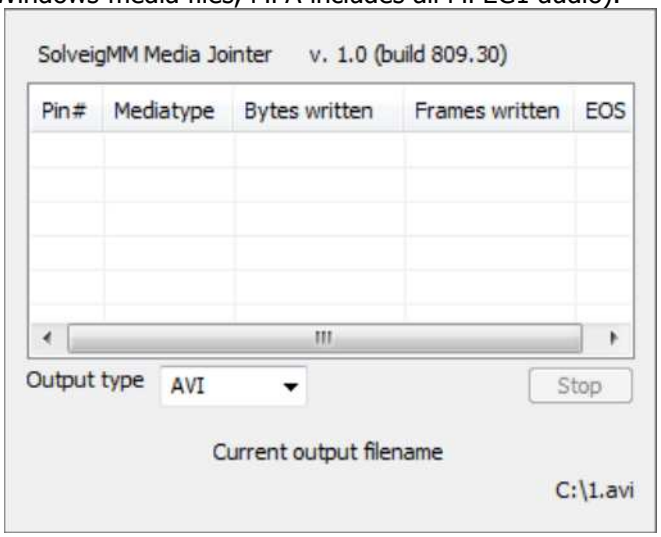


Fig 3

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:

Parameter GUID	**Value type**	**Description**
SMMJ_FileName	VT_BSTR	Set/retrieve the output file's name
SMMJ_BytesWrittenByPin	in - VT_INT	Retrieve the number of bytes written by specified pin

	out - VT_UI8	
SMMJ_GetFullStatistic	VT_BYREF |VT_UI1	Retrieves the full writing statistic of each pin. (See below)
SMMJ_StopWriting	-	
SMMJ_FileType	VT_INT	Set/retrieve the output file's type.
SMMJ_AvgTimeWritten	VT_I8	Retrieves the current duration of the output file.
SMMJ_ExternalMessageWindow	-	Set the external window to receive the graph messages.

SMMJ_FileName

GUID:
Available in a FULL version of SDK

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

Type:
VT_BSTR (bstrVal)

SMMJ_BytesWrittenByPin

GUID:
Available in a FULL version of SDK

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:

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

SMMJ_FileType

GUID:
Available in a FULL version of SDK

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:
Available in a FULL version of SDK

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

Type:
VT_I8 (llVal)

SMMJ_ExternalMessageWindow

GUID:
Available in a FULL version of SDK

Description:
Specifies the external window to receive graph messages.

Type:
not applied (pvRecord)

SMMJ_GetFullStatistic

GUID:
Available in a FULL version of SDK

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

```
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

How to perform 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 ITrimmerObjectControlCB 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/end 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.
  wfName - the name of the file to be appended.

  This step is shown in the GetGroupTaskFromList() member function.

- **Set the folllowing 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 compatibilit**y (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.

SDK Batch file structure

Batch files to be supported by SolveigMM Video Editing SDK are Extensible Markup Language (XML) with extension - *.xtl and structure based on DirectShow Editing Services XTL format. This section documents the XTL elements and attributes.

> ***Note:*** *All the elements and attributes are sensitive to keyboard case and must be in lower case. The values must be quoted ("").*

This table contains the basic elements of the XTL file:

Element	Description
clip	Specifies a media source file
group	Defines a group, the top-level object in a timeline. Specifies output file name
timeline	Defines a timeline. This element is the root node in the XTL file
track	Defines a track object (media files)

Here listed all supported attributes:

Attribute	Possible value	Obligation presence	Belonging to element	Description
name	Any string excepting symbol (")	Yes	group	Specifies an output file name with a path
video	[0, n]	No	track	Specifies a quantity of video streams
audio	[0, n]	No	track	Specifies a quantity of audio streams
obey_sample_time	[0, 1]	No	track	Specifies to use an alternative synchronization algorithm
out_type	[avi, asf, mpg_ves, mpg_ps, mpg_ts]	No	track	Specifies an output file format
src	Any string excepting symbol (")	Yes	clip	Specifies an input file name with a path
start	HH:MM:SS.mS	Yes	clip	Specifies the start time of a fragment to be saved into an output file
stop	HH:MM:SS.mS	Yes	clip	Specifies the stop time of a fragment to be saved into an output file

timeline Element

The timeline element defines the timeline. This element is the root node in the XML file.

Example:

```
<timeline></timeline>
```

Attributes:

no attributes specified

Parent/Child information

Parent	Children
None, this is the root element	group

group Element

The group element defines a group. The top level object is a timeline.

Example:

```
<group name="c:\temp\test_output.wmv"></group>
```

Attributes

Attribute	Pssible values	Obligation presence	Description
name	Any string, except (") symbol	Yes	Specifies an output filename with a path

Parent/Child information

Parent	Children
timeline	track

track Element

The track element defines an output file.

Example:

```
<track video="1" audio= "1" obey_sample_times= "0"></track>
```

Attributes

Attribute	Pssible values	Obligation presence	Description
video	[0, n]	No	Specifies a quantity of video streams
audio	[0, n]	No	Specifies a quantity of audio streams
obey_sample_time	[0, 1]	No	Specifies to use an alternative synchronization algorithm
out_type	[avi, asf, mpg_ves, mpg_ps, mpg_ts]	No	Specifies an output filename with a path

Parent/Child information

Parent	Children
group	clip

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.

Example:

```
<clip src="C:\video\test_input.wmv" start="00:32:41.00" stop= "00:42:02.00"/>
```

Attributes

Attribute	Pssible values	Obligation presence	Description
src	Any string, except (") symbol	Yes	Specifies an input file name with a path
start	HH:MM:SS:mS	Yes	Specifies the start time of a fragment to be saved into an output file
stop	HH:MM:SS:mS	Yes	Specifies the stop time of a fragment to be saved into an output file

Parent/Child information

Parent	Children
track	None

Batch file example:

```
<timeline>
    <group name="C:\video\ResultFile_1.avi">
        <track video="1" audio="2" obey_sample_times="0">
            <clip src="C:\video\Progulka.avi" start="00:00:00.00" stop="00:12:32.00" />
            <clip src="C:\video\Progulka.avi" start="00:32:41.00" stop="00:42:02.00" />
            <clip src="C:\video\Progulka.avi" start="00:54:09.00" stop="01:08:51.00" />
        </track>
    </group>
    <group name="C:\video\ResultFile_2.avi">
        <track video="0" audio="1" obey_sample_times="1">
            <clip src="C:\video\Micro.avi" start="00:11:48:08" stop="00:20:14.52 /">
        </track>
    </group>
</timeline>
```

Remarks:
As a result there should be created two files:

- C:\video\ResultFile_1.avi, the file would consist of three sequential fragments from input file C:\video\Progulka.avi and has first video stream and second audio streams of input file
- C:\video\ResultFile_2.avi, the file would consist of one fragment from input file C:\video\Micro.avi and has only first audio stream of an input file.