

SolveigMM Video Editing SDK x64

Developer Reference Manual

SDK Version: 3.0 FULL

First edition: February 27, 2007

Date modified: April 1, 2016

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

Contents

SolveigMM Video Editing SDK Notices	4
Release Notes.....	5
New features in SolveigMM Video Editing SDK x64 3.0	5
Product Description	5
Components.....	6
Sample applications	7
System requirements.....	8
User Guide	9
Installation/Uninstallation.....	9
SDK folder structure	9
SDK Sample Applications	9
SolveigMM Video Editing SDK Reference Manual	10
SolveigMM Video Editing Engine	10
Editing Engine enumerations	10
SMM_TaskType.....	10
SMM_TrimAccuracyType.....	11
SMM_TrimFlags	11
Editing Engine structures	13
SMM_TRACK_INFO.....	13
SMM_CLIP_INFO.....	15
TrimInfoList.....	16
FILE_VERSIONS_2	17
VERSION_INFO_2	18
SMM_ASF_MARKER.....	18
Editing Engine Interfaces	21
ITrimmerObjControl interface	22
ITrimmerObjControl::Start	23
ITrimmerObjControl::Stop.....	24
ITrimmerObjControlCB interface	24
Editing Engine Parameters	26
SMAT_InputFile	27
SMAT_OutputFile	28
SMAT_Flags.....	28
SMAT_ResetFlags.....	28
SMAT_TrimList.....	28
SMAT_Progress.....	29
SMAT_AudioStreamNumber.....	29
SMAT_VideoStreamNumber.....	29
SMAT_Callback	30
SMAT_GetVersions2	30
SMAT_CurrentSize.....	31
SMAT_BatchFileName	31
SMAT_CurrentBatchTask.....	31
SMAT_OutputType	31
SMAT_BatchTasksNum	32
SMAT_TaskType	32
SMAT_TrackInfo	32
SMAT_MPEG2IndexFileName	32
SMAT_SilentPath.....	33
SMAT_MPEG2CodecsSet	33
SMAT_CodecsSet	33
SMAT_GetFileType	34
SMAT_ValidateFiles	34

Required steps before start joining	36
SMAT_ASFMarker	36
SMAT_ASFMarkersCount	37
SMAT_ASFMarkerRemove	37
SolveigMM Media Joiner Filter	38
Requirements	38
Usage	39
SolveigMM MP4 Muxer filter	43
Features	43
Usage	43
SolveigMM MP4 Splitter filter	46
Features	46
Usage	46
SolveigMM AVC Video Trimmer FA filter	48
SolveigMM Silence Generator Filter	49
Features	49
SDK Batch Processing	51
SDK Batch file structure	51
Batch samples	55
Batch trimming	56
Batch joining	56
Batch indexation	57
Batch extract video	57
Batch extract audio	58

SolveigMM Video Editing SDK Notices

SDK Version: 3.0 FULL
First edition: February 27, 2007
Date modified: April 1, 2016

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 3, 634055,
Tomsk, Russian Federation
www.solveigmm.com

© 2006 – 2016 Solveig Multimedia. All rights reserved.

Release Notes

New features in SolveigMM Video Editing SDK x64 3.0

Multiple files trimming for the MP4 format

The feature allows developers to create an environment where one can select fragments from several different MP4 files with H.264/AVC video with identical parameters (video/audio content, bit rate, width, height) and save them into a single file

The components that provide the feature are as follows:

- SolveigMM AVC Video Trimmer Filter,
- SolveigMM Audio Trimmer Filter,
- SolveigMM Trimm Graph Manager,
- SolveigMM Timeline Editing Object.

For more information see [Components](#) description and [SolveigMM Batch Splitter Sample Application](#)

Product Description

The SolveigMM Video Editing SDK x64 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**
 - MP4 format (MPEG-4 Part 14);
 - MOV format (QuickTime multimedia file format);
- **Multipart frame accurate trimming**
 - MP4 format (MPEG-4 Part 14);
 - MOV format (QuickTime multimedia file format);
- **Media files joining**
 - MP4 format (MPEG-4 Part 14);
 - MOV format (QuickTime multimedia file format);
- **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)**

Components

SolveigMM Video SDK x64 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 MP4, MOV files without decoding/encoding.
- **SolveigMM Custom Encoder (SMM_CustomEncoder.dll)**
Library for partial reencoding of video fragments.
- **SolveigMM Audio Silence Generation Library (SMM_SilenceGen.ax)**
COM object for getting samples for muted audio with different compression format.

DirectShow Trimmer filters:

- **SolveigMM Audio Trimmer (SMM_AudioTrimmer.ax)**
DirectShow filter for GOP accurate audio streams trimming.
- **SolveigMM AVC Frame Accurate Trimmer (SMM_AVCTrimmer.ax)**
DirectShow filter for AVC/H264 video streams 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 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.

Other DirectShow filters:

- **SolveigMM Media Joiner (SMM_MediaJoiner.ax)**
DirectShow filter for joining video/audio streams of different files. Currently supported mode: whole file-by-file joining. 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 Timeline Editing Object (SMM_TimelineEditingObj.ax)**
DirectShow filter for editing tasks which are set as XTL files.
For more information on possible XTL file format see [SDK Batch file structure](#).
- **SolveigMM Trimm Graph Manager (SMM_TrimmGraphManager.ax)**
DirectShow filter for managing trimming filters.

Sample applications

C++ Samples

Name of Sample	Description
SMM 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.

System requirements

- CPU (Intel® Pentium II, Celeron, AMD® Athlon, Opteron etc.)
- 128 MB RAM.
- Any VGA card.
- Windows® XP / 2000 / 2003 / Vista / 7 / 8 / 10
- 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)

User Guide

Installation/Uninstallation

Installing SDK

1. To install the SolveigMM Video Editing SDK x64:
2. Run the SDK setup. To run, double click the executable file from the SDK setup package.
3. The SolveigMM Video Editing SDK x64 window will appear. Read the recommendations and warnings. Click Next.
4. The Release Notes will appear. 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 x64 has been successfully installed" will appear and the SDK is ready to use.

Uninstalling SDK

To uninstall the SolveigMM Video Editing SDK x64:

1. Click -> Start -> Programs -> Solveig Multimedia -> SolveigMM Video Editing SDK x64 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 folder structure

After installing the SDK, the "SolveigMM Video Editing SDK x64" 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 x64** 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

Samples – includes VC++ Sample Application and media files

SDK Sample Applications

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 Video Editing SDK Reference Manual

SolveigMM Video Editing Engine

SolveigMM Video Editing Engine COM is the basic SDK's object. It 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 of the Video Editing SDK x64 allows frame/GOP accurate trimming and joining of MP4, MOV files.

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.

SMM_TRACK_INFO(Class)

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 curent clip.

rtStart

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

rtEnd

End time of the curent 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 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 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 x64.

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

Interface	Description
IUnknown	
QueryInterface	Retrieves pointers to supported interfaces
AddRef	Increments reference count
Release	Decrements reference count
IPersist	
GetClassID	Retrieves the class identifier (CLSID) of an object
IPersistStream	
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	

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

Interface	Description
IUnknown	
QueryInterface	Retrieves pointers to supported interfaces
AddRef	Increments reference count

Release	Decrements reference count
IDispatch	
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	
Start	Starts the engine process
Stop	Stops the engine 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

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

The Stop method stops the current engine 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

Interface	Description
IUnknown	
QueryInterface	Retrieves pointers to supported interfaces
AddRef	Increments reference count
Release	Decrements reference count
ITrimmerObjControlCB	
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

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

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

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

Editing Engine Parameters

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.

Parameter GUID	Value type	Available values	Default	Description
SMAT_InputFile	VT_BSTR		-	Specifies an input media file name
SMAT_OutputFile	VT_BSTR		-	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 via TrimInfoList structure
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_OutputType	VT_UI4	[SMM_OutputTypes]	input	Specifies an output file type

			type	
SMAT_BatchTasksNum	VT_UI4	Read only	-	Retrieves a count of tasks a set batch file contains
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 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	Should be replaced with SMAT_CodecsSet. Sets the set of codecs to use for MPEG2 files editing.
SMAT_CodecsSet	VT_BSTR		See below	Sets the set of codecs to use for 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:

{4890910-6B0B-465D-B507-2568710565FD}

Description:

Specifies an input media file name. For details please see SMM Trim C++ [BatchSplit](#) sample application.

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++ [BatchSplit](#) sample applications.

Type:
VT_BSTR

SMAT_Flags

GUID:
{AAF40DFF-0698-4424-87DC-3B1E2D60CDD8}

Description:
Set trimming configuration flags to the Video Editing Engine.

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.

Type:
VT_UI4

SMAT_TrimList

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

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.

Type:

VT_BYREF | VT_UI1

SMAT_Progress

GUID:

{AE82DE01-D84B-4EE5-90AA-052B108C4380}

Description:

Retrieves a progress of trimming process in percentage.

Type:

VT_I4

Available Values:

0 – 100 (%)

SMAT_AudioStreamNumber

GUID:

{5EB26970-133F-4ad1-BE21-A849618CA8F3}

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:

{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](#) sample application.

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.lVal;
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:**

{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](#) paramater

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-52D9-4f41-9091-11C4FDAE990B}

Type:

VT_UI4

Description:

Specifies which process is to be performed with chosen files(See [SMM TaskType](#)).

Available values:

see [Editing Engine enumerations](#).

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 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-5045-4732-436F6E666967}

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:

```
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:

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

Type:

Input - not applied;
Output - VT_I4(IVal);

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->GetValue( &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 [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:

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 IVal to HRESULT.

If you check the result with "SUCCEEDED()" macros remember that it returns **TRUE** if HRESULT(IVal) == **S_FALSE** and **FALSE** if HRESULT(IVal) == **E_FAIL** or other errors. You can avoid using this automatic validation due to absence 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/end of the interval of file to be joined. Currently this 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 these 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 Media Joiner Filter

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

Filter Interfaces	IBaseFilter, IFileSinkFilter, IModuleConfig, IAMFilterMiscFlags, ISpecifyPropertyPages
Input Pin Media Types	MEDIATYPE_Audio, MEDIATYPE_Video, MEDIASUBTYPE_NULL
Input Pin Interfaces	IPin, IMemInputPin
Output Pin Media Types	Not applicable.
Output Pin Interfaces	Not applicable.
Filter CLSID	CLSID_SMM_Jointer
Property Page CLSID	CLSID_SMM_Jointer_Prop_Main
Executable	SMM_MediaJoiner.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory
Declared in	PropID_MediaJoiner.h

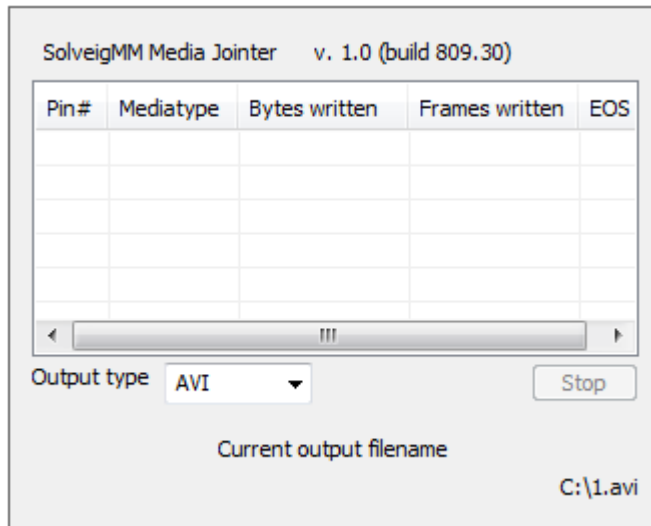
Requirements

The filter currently supports the following formats: MP4, MOV. 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.



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 out - VT_UI8	Retrieve the number of bytes written by specified pin
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	Retrieve the current duration of the output file.

SMMJ_ExternalMessageWindow	-	Set the external window to receive the graph messages.
----------------------------	---	--

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:

{534d4d4a-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:

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

SMMJ_FileType

GUID:

{534d4d4a-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:

{534d4d4a-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

```
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

lFramesCount

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 MP4 Muxer filter

The MP4/MOV muxer filter implemented as a DirectShow filter. Its purpose is to multiplex input video/audio/subtitles into MP4 container.

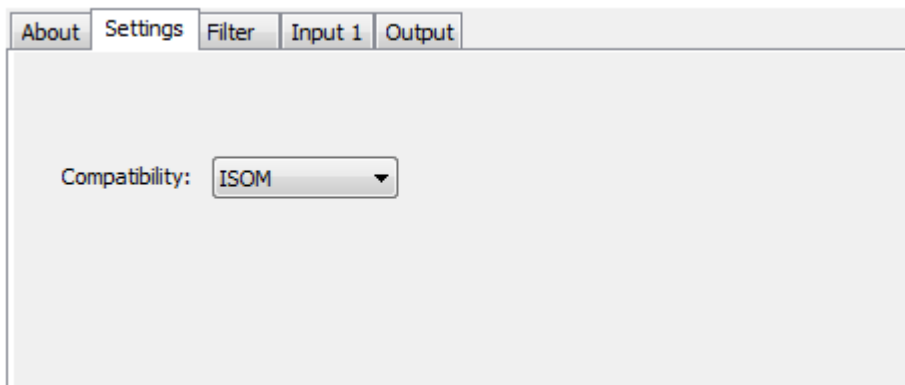
Filter Interfaces	IBaseFilter, IModuleConfig, ISpecifyPropertyPages
Input Pin Media Types	MEDIATYPE_Video, MEDIASUBTYPE_NULL MEDIATYPE_Audio, MEDIASUBTYPE_NULL
Input Pin Interfaces	IPin, IMemInputPin
Output Pin Media Types	MEDIATYPE_Stream, MEDIASUBTYPE_NULL
Output Pin Interfaces	IPin, IMemInputPin
Filter CLSID	CLSID_SMM_MP4_MUXER
Property Page CLSID	Properties - {D10A6FDF-9EDB-44b5-A3EA-5AFE45E0FCA}, About - {A8314949-4BA9-4a2c-9E76-EAB7FB9C5BF5}
Executable	SMM_MP4Muxer.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory
Declared in	PropID_MP4Muxer.h

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 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:

Parameter GUID	Value type	Description
SMMP4M_MUX_TIME	VT_I8	Reads the current muxing time
SMMP4_MUX_TYPE	VT_I4	Specifies an output stream type.

SMMP4M_MUX_TIME

GUID:

{5C99C340-3DC0-43c3-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

```
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.

MMT_3gp4

Stream type is.

MMT_avc1

Stream type is .

MMT_mux_type_cnt

Elements number.

SolveigMM MP4 Splitter filter

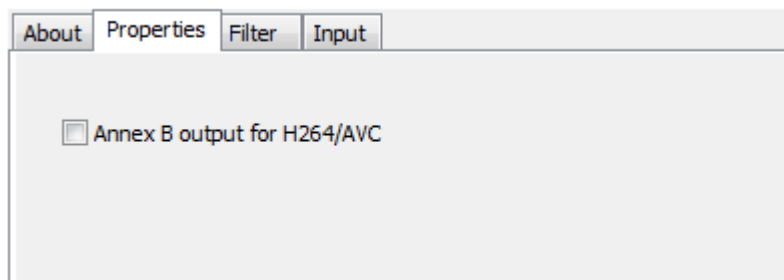
The MP4 splitter filter implemented as a DirectShow filter. It can parse MP4 or MOV files

Filter Interfaces	IBaseFilter, IModuleConfig, ISpecifyPropertyPages
Input Pin Media Types	MEDIATYPE_Stream, MEDIASUBTYPE_NULL
Input Pin Interfaces	IPin, IMemInputPin
Output Pin Media Types	MEDIATYPE_Video, MEDIASUBTYPE_NULL; MEDIATYPE_Audio, MEDIASUBTYPE_NULL;
Output Pin Interfaces	IPin, IMemInputPin
Filter CLSID	CLSID_SMM_MP4_DEMUXER
Property Page CLSID	Properties - {0794A19A-4EB1-4cf9-A25B-5B4004423751}, About - {174B905B-CFF5-4913-8302-4244F183779F}
Executable	SMM_MP4Muxer.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory
Declared in	PropID_MP4Demuxer.h

Features

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

- Annex B Output checkbox - Enables the AnnexB output stream's format for AVC/H264 video.



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_MP4Demuxer.h header file 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:

Parameter GUID	Value type	Description
SMMP4_H264_ANNEXB	VT_BOOL	Sets/Gets the compatibility with AVC ANNEX B for AVC/H264 video
SMMP4_MUX_TYPE	VT_I4	Gets current filte's type.

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 AVC Video Trimmer FA filter

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

Filter Interfaces	IBaseFilter, IModuleConfig, ISpecifyPropertyPages
Input Pin Media Types	MEDIATYPE_Video, MEDIASUBTYPE_MPEG2_VIDEO MEDIASUBTYPE_NULL
Input Pin Interfaces	IPin, IMemInputPin
Output Pin Media Types	MEDIATYPE_Video, MEDIASUBTYPE_MPEG2_VIDEO MEDIASUBTYPE_NULL
Output Pin Interfaces	IPin, IMemInputPin
Filter CLSID	CLSID_AVCVideoTrimmerFilter
Property Page CLSID	CLSID_AVCVideoTrimmerFilter_PPage, CLSID_AVCVideoTrimmerFilter_About_PPage
Executable	SMM_AVCVideoTrimmer.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory
Declared in	PropID_AVCVideoTrimmer.h

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.

Filter Interfaces	IBaseFilter, IModuleConfig, ISpecifyPropertyPages
Filter CLSID	CLSID_SMSilenceGen
Executable	SMM_SilenceGen.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory
Declared in	PropID_SilenceGen.h

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:

Parameter GUID	Value type	Description
SMSG_Init	VT_BYREF VT_UI1	Initialize by SgInitStruct
SMSG_SilenceSampleSize	VT_UI4	Get Silence Sample size
SMSG_SilenceSample	VT_BYREF VT_UI1	Get Silence Sample data (buffer should be preallocated)
SMSG_SliceSampleDuration	VT_I8	Get Silence Sample Duration

SMSG_Init

GUID:

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

Description:

Initialize by SgInitStruct

SgInitStruct

```
struct SgInitStruct
{
    BSTR silenceBaseXMLFile;
```

```
BSTR filePathName;  
int streamNum; //1 base  
SMM_OutputTypes fileType;  
BSTR profileString;  
BSTR silentPath;  
};
```

Type:

VT_BYREF | VT_UI1 (pbVal)

MSG_SilenceSampleSize**GUID:**

{A1683E2B-4508-4dfa-AAE6-7D9B30CC231F}

Description:

Get Silence Sample size

Type:

VT_UI4(IVal)

MSG_SilenceSample**GUID:**

{41B87714-AD4E-4d9a-B13E-A76C23316065}

Description:

Get Silence Sample data (buffer should be preallocated)

Type:

VT_BYREF | VT_UI1 (pbVal)

MSG_SliceSampleDuration**GUID:**

{3D911100-8A9B-46e8-A690-4EEE1007EA2E}

Description:

Get Silence Sample Duration

Type:

VT_I8 (IVal)

SDK Batch Processing

SDK Batch file structure

Batch files to be supported by SolveigMM Video Editing SDK x64 are Extensible Markup Language (XML) with *.xfl extension and structure based on Direct Show Editing Services XFL format. This section documents the XFL 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 XFL 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 can be the root node in the XFL file
timelines	Joins several timelines. This element is the root node in the XFL file
track	Defines a track object (media files)

Here listed all supported attributes:

Attribute	Possible value	Obligation presence	Belonging to element	Description
output	Any string excepting symbol ("")	Yes	group	Specifies an output file name with a path
out_type	[avi, asf, mpa, mpg_ps, mpg_ts, mpg_ves, mpg1_sys, mpg1_ves, wav, avc_ves, matroska, mp4]	No	group	Specifies an output file's type. If not specified - the same as input file.
mode	[trimming, joining, indexation, multiplexing]	No	group	Specifies specifies the operation type of current task. If not specified - trimming
video	[-1, n]	No	track	Specifies stream number of video ("-1" - means all streams, "0"

				- means disable stream)
audio	[-1, n]	No	track	Specifies stream number of audio ("-1" - means all streams, "0" - means disable stream)
accuracy	[gop, frame]	No	track	Specifies a trimming accuracy. If not specified - the frame accuracy is used if it is implemented (see frame accurate trimming formats) else used GOP accuracy mode.
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
timeFormat	[100ns_units, time10ms, position]	No	clip	Specifies the format of "start" and "stop" attributes. If not specified - time10ms
start	HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units "begin" keyword - if timeFormat is position	Yes	clip	Specifies the start time of a fragment to be saved into an output file
stop	HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units "end" keyword - if timeFormat is position	Yes	clip	Specifies the stop time of a fragment to be saved into an output file

timelines Element

The timelines element joins several timelines. This element is the root node in the XML file.

Attributes

Attribute	Possible values	Obligation presence	Description
version	2	No	Specifies the XTL format version

Parent/Child information

Parent	Children
None, this is the root element	timeline

timeline Element

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

Attributes:

no attributes specified

Parent/Child information

Parent	Children
timelines	group

group Element

The group element defines a group. The top level object is a [timeline](#).

Attributes

Attribute	Possible values	Obligation presence	Description
output	Any string, except (") symbol	Yes	Specifies an output filename with a path
mode	[trimming, joining, indexation, multiplexing]	No	Specifies specifies the operation type of current task. Default value - trimming

Parent/Child information

Parent	Children
timeline	track

track Element

The track element defines an output file.

Attributes

Attribute	Possible values	Obligation presence	Description
video	[-1, n]	No	Specifies stream number of video ("-1" - means all streams, "0" - means disable stream)
audio	[-1, n]	No	Specifies stream number of audio ("-1" - means all streams, "0" - means disable stream)
obey_sample_time	[0, 1]	No	Specifies to use an alternative synchronization algorithm
out_type	[avi, asf, mpa, mpg_ps, mpg_ts, mpg_ves, mpg1_sys, mpg1_ves, wav, avc_ves, matroska, mp4]	No	Specifies an output file's type. If not specified - the same as input file.
accuracy	[gop, frame]	No	Specifies a trimming accuracy. Default value - the best available accuracy type for current format.

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 to 1. 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
start	HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units "begin" keyword - if timeFormat is position	Yes	Specifies the start time of a fragment to be saved into an output file
stop	HH:MM:SS.mS - if timeFormat is time10ms 100ns - if timeFormat is 100ns_units "end" keyword - if timeFormat is position	Yes	Specifies the stop time of a fragment to be saved into an output file
timeFormat	[100ns_units, time10ms, position]	No	Specifies the format of "start" and "stop" attributes. If not specified - time10ms
flags	"audio_silence"	No	Specifies that audio in the segment will be replaced by silence in the resulted file

Parent/Child information

Parent	Children
track	None

Batch samples

All XTL samples located in C:\Program Files\Solveig Multimedia\SolveigMM Video Editing SDK x64\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
x64\Samples\Media\Batch XTL Samples\Batch trimming.xml"
```

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:

Samples\Media\Batch XTL Samples\Batch trimming.xml

```
<timelines version="2">
  <timeline>
    <group output="..\output\sample_mp4_0001.mp4">
      <track video="1" audio="1" accuracy="frame">
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:01:23" stop="00:00:03:50"
timeFormat="time10ms"/>
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:07:28" stop="00:00:12:52"
timeFormat="time10ms"/>
      </track>
    </group>
  </timeline>
</timelines>
```

Remarks:

The most important attributes here are:

output - output file,

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_mp4_0001.mp4 should be created

Batch joining:

Samples\Media\Batch XTL Samples\Batch joining.xml

```
<timelines version="2">
  <timeline>
    <group output="..\output\sample_mp4_Joined.mp4" out_type="mp4">
      <track video="-1" audio="-1">
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:00:00" stop="00:00:15:12"
timeFormat="time10ms"/>
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:00:00" stop="00:00:15:12"
timeFormat="time10ms"/>
      </track>
    </group>
  </timeline>
</timelines>
```



```
timeFormat="time10ms"/>
    </track>
  </group>
</timeline>
</timelines>
```

Remarks:

The most important attributes here are:

output - output file,

out_type - the type of output file,

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_mp4_Joined.mp4 should be created

Batch indexation:

Samples\Media\Batch XTL Samples\Batch indexation.xtl

```
<timelines version="2">
  <timeline>
    <group output="..\mpeg1-ss\sample_mpeg1_sys.mpg.mlmix" mode="indexation">
      <track>
        <clip src="..\mpeg1-ss\sample_mpeg1_sys.mpg" start="00:00:00:00"
stop="00:00:00:00"/>
      </track>
    </group>
  </timeline>
</timelines>
```

Remarks:

The most important attributes here are:

output - 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:

Samples\Media\Batch XTL Samples\Batch extract video.xtl

```
<timelines version="2">
  <timeline>
```

```

    <group output="..\output\sample_mp4_ves_0001.h264" out_type="avc_ves">
      <track video="1" audio="0" accuracy="frame">
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:00:00" stop="00:00:15:92"
timeFormat="time10ms"/>
      </track>
    </group>
  </timeline>
</timelines>

```

Remarks:

This operation is only supported for mpeg1-mpeg2 files.
The most important attributes here are the same as for trimming, except:
audio - should always be set to 0,
out_type - should be set to mpeg_ves
start, stop - should be set to 0 and a full length of file accordingly.

As a result the file sample_mpeg2_ts_0001.mpg should be created

Batch extract audio:

Samples\Media\Batch XTL Samples\Batch extract audio.xtl

```

<timelines version="2">
  <timeline>
    <group output="..\output\sample_mp4_0001.aac" out_type="raw">
      <track video="0" audio="1" accuracy="gop">
        <clip src="..\mp4\sample_mp4.mp4" start="00:00:00:00" stop="00:00:15:92"/>
      </track>
    </group>
  </timeline>
</timelines>

```

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