



**SolveigMultimedia**  
software that works

# **SolveigMM MPEG-2 Requantizer SDK**

## **Developer Reference Manual**

SDK Version: 1.0

First edition: July 10, 2006

Date modified: February 12, 2007

Solveig Multimedia,  
Acadimicheskij 10/3, 634055,  
Tomsk, Russian Federation  
[www.solveigmm.com](http://www.solveigmm.com)

## Table of contents

1	<a href="#">SolveigMM MPEG-2 Requantizer SDK Notices</a>	3
2	<a href="#">Releases Notes</a>	4
2.1	<a href="#">Product Description</a>	4
2.2	<a href="#">Components</a>	4
2.4	<a href="#">Samples</a>	4
2.5	<a href="#">System Requirements</a>	4
3	<a href="#">SDK User Guide</a>	5
3.1	<a href="#">Installing SDK</a>	5
3.2	<a href="#">Uninstalling SDK</a>	5
3.3	<a href="#">SDK Folder Structure</a>	5
3.4	<a href="#">SDK Sample Applications</a>	6
4	<a href="#">SDK Reference Manual</a>	9

## **SolveigMM MPEG-2 Requantizer SDK Notices**

SDK Version: 1.0

First edition: July 10, 2006

Date modified: February 12, 2007

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

For Technical Support, contact the Solveig Multimedia Technical Support Team: [support@solveigmm.com](mailto:support@solveigmm.com)

For purchasing and licensing information, contact the Sales Department: [info@solveigmm.com](mailto: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

### PRODUCT DESCRIPTION

The SolveigMM MPEG-2 Requantizer SDK is a software development kit to enable programmers to use SolveigMM MPEG-2 Requantizer component in digital video/audio applications for a task of swift and quality MPEG-2 high bitrate video streams transcoding.

### COMPONENTS

SolveigMM MPEG-2 Requantizer SDK involves **SolveigMM MPEG-2 Requantizer** DirectShow ® filter (**SMM\_MP2Req.ax**) to implement the smart technology of MPEG-2 Video (ISO/IEC 13818-2) fast recompression (requantizing / transrating / transcoding).

### SAMPLES

#### RequantDemo

This sample demonstrates using of SolveigMM MPEG-2 Requantizer DirectShow ® filter. The application inputs are MPEG-2 Video Elementary Stream, MPEG-2 Program Stream, DVD VOB files. It splits video/audio streams, recompresses MPEG-2 VES in concordance with the set compression ratio and writes audio/video streams to respective files. Audio streams are dumped as is, without recompression. Total audio/video files size is approximate size of MPEG-2 or VOB file containing these streams after multiplexing. Such one can estimate potential VOB size decrease.

### 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.

## SolveigMM MPEG-2 Requantizer SDK User Guide

### Installing SDK

To install the SolveigMM MPEG-2 Requantizer SDK:

1. Run the SDK setup. To run, double click the executable file from the SDK setup package.
2. The *SolveigMM MPEG-2 Requantizer 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 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 MPEG-2 Requantizer SDK has been successfully installed" will appear and the SDK is ready to use.

### Uninstalling SDK

To uninstall the SolveigMM MPEG-2 Requantizer SDK

1. Click *Start*→*Programs*→*Solveig Multimedia*→*SolveigMM MPEG-2 Requantizer SDK xx* →*Uninstall SDK* (xx – the SDK version number).

Follow the onscreen instructions to complete removal of the application.

### SDK Folder Structure

After installing the SDK, the "SolveigMM MPEG-2 Requantizer SDK" folder will appear in the destination folder specified during installation.

---

**Note:** After installing the SolveigMM MPEG-2 Requantizer 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 MPEG-2 Requantizer 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\Moonlight\_Headers\Inc\**
- **Include\Moonlight\_Headers\Classes\** - includes MONLIGHT SDK headers to be used only by the sample for creating DirectShow filter graph

**Lib** - includes MONLIGHT SDK debug\release libraries to be used only by the sample for creating DirectShow filter graph

**Samples** – includes VC++ 7.0 Sample Application

### SDK Sample Applications

#### RequantDemo

This sample demonstrates using of SolveigMM MPEG-2 Requantizer DirectShow ® filter. The application inputs are MPEG-2 Video Elementary Stream, MPEG-2 Program Stream, DVD VOB files. It splits video/audio streams, recompresses MPEG-2 VES in concordance with the set compression ratio and writes audio/video streams to respective files. Audio streams are dumped as is, without recompression. Total audio/video files size is approximate size of MPEG-2 or VOB file containing these streams after multiplexing. Such one can estimate potential VOB size decrease.

The SolveigMM MPEG-2 Requantizer DirectShow filter can accept advanced parameters (Figure 2). It is possible to be set a unique compression rate for each picture type (Intra, Predicted or Bi-directional) and each of macroblock types (intra and non-intra). For more information about parameters see [put AdvSettings](#) and [put CompRatio](#) remark sections.

The application (Figure 1) builds a DirectShow graph depicted on Figure 3, 4, and allows the user to set the compression ratio (Figure 2) and start/stop recompression process.

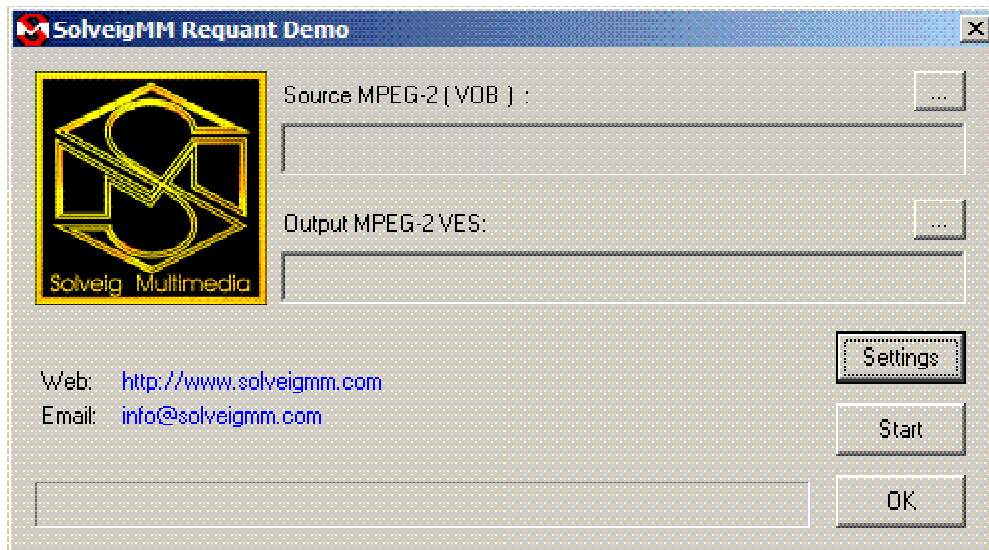


Figure 1 – The application GUI

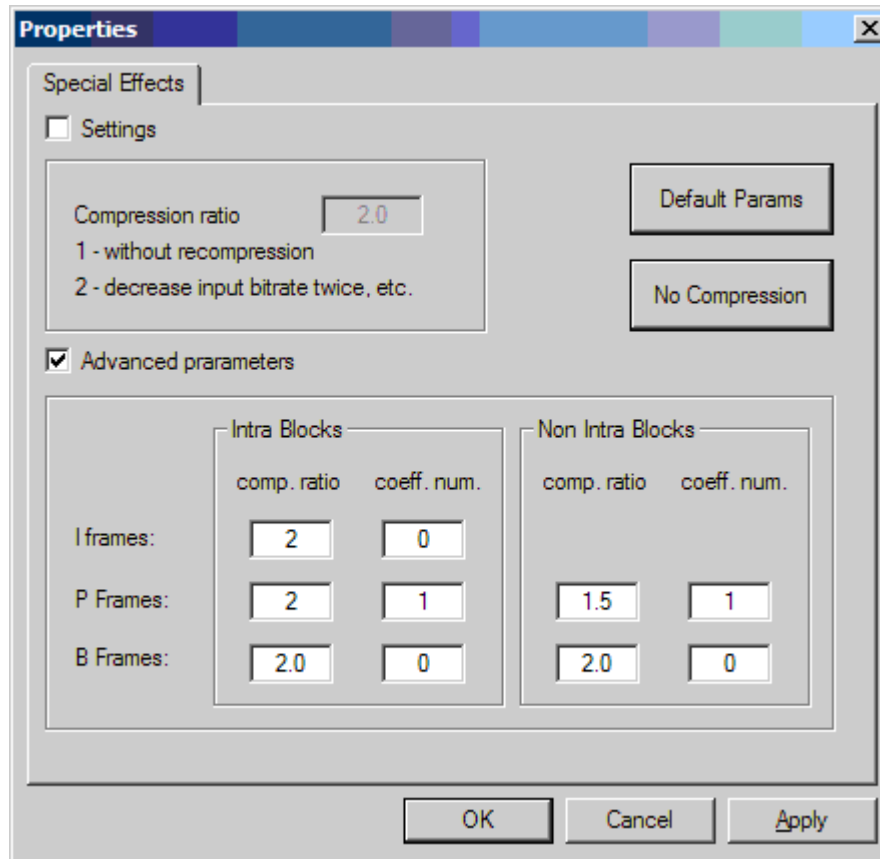


Figure 2 – The SolveigMM MPEG-2 Requantizer settings

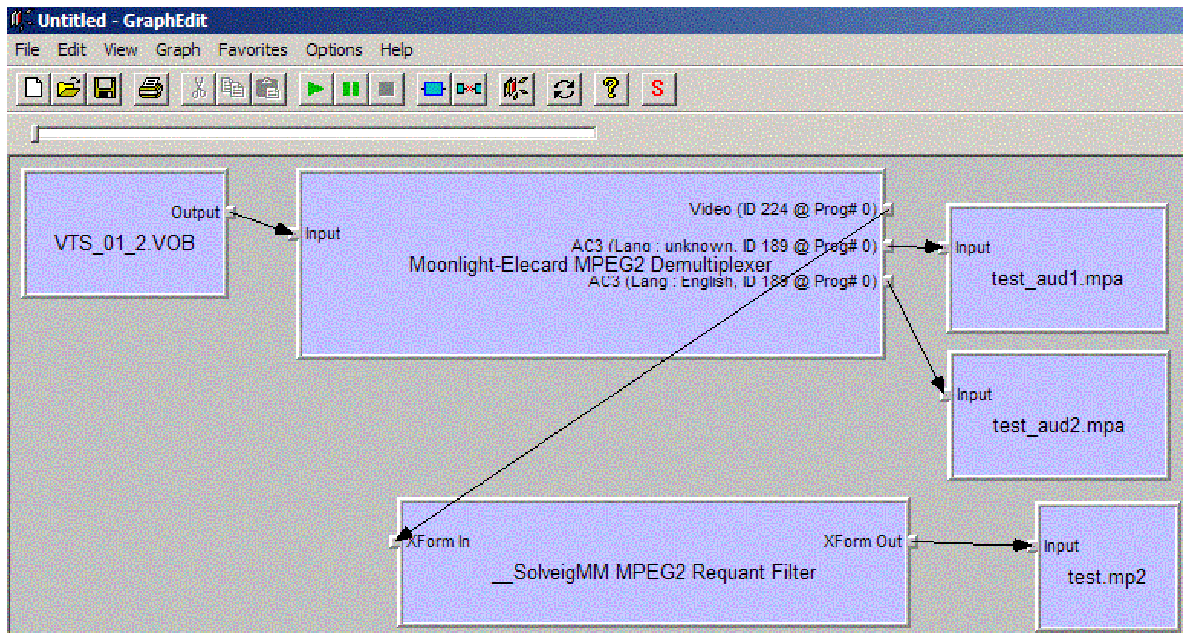
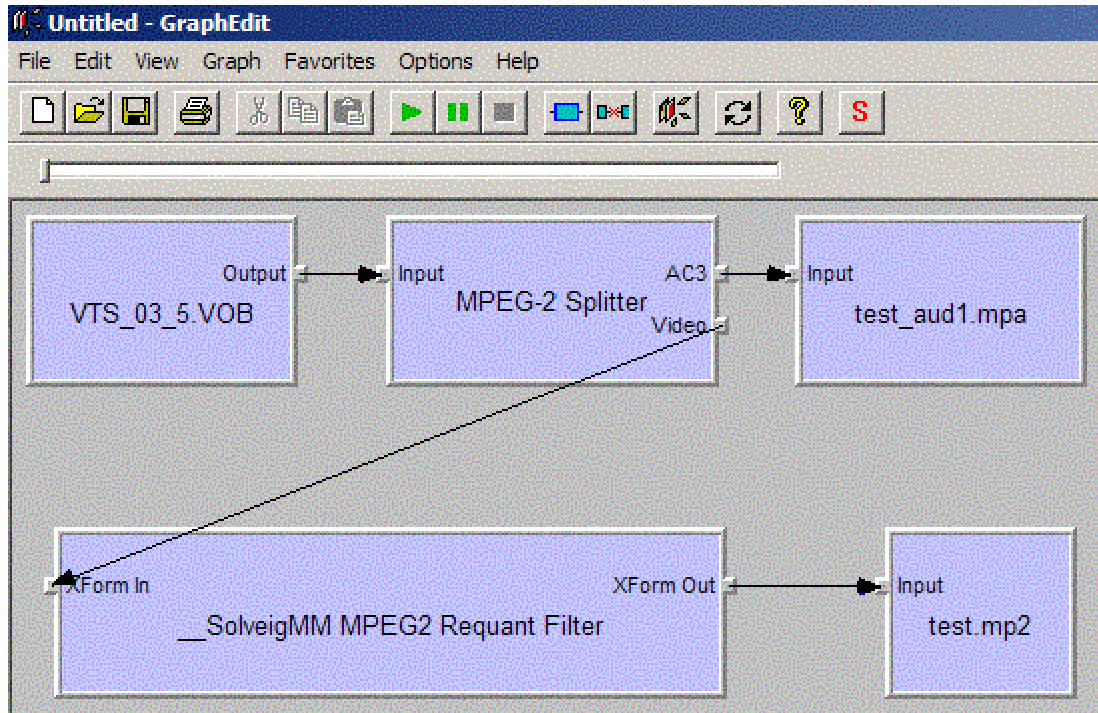


Figure 3 –Recompression graph with Elecard MPEG2 Demultiplexer

---

## SolveigMM MPEG-2 Requantizer SDK



Picture 4 – Recompression graph with Microsoft MPEG2 Demultiplexer



## SolveigMM MPEG-2 Requantizing SDK Reference Manual

**SolveigMM MPEG-2 Requant** filter is a software-only requantizing (recompression / transrating / transcoding) solution for MPEG-2 video (ISO/IEC 13818-2). The engine is implemented as a DirectShow® filter and can be used by any DirectShow® application. The filter can process MPEG-2 video streams produced by the Elecard MPEG Demultiplexer filter or by Microsoft® MPEG-2 Splitter. The filter output pin can be connected to any DirectShow® filter to accept MPEG-2 video data, such MPEG-2 Multiplexer, Dump filter, Net Renderer, MPEG-2 Video Decoder and so forth.

Filter Interfaces	<b>IBaseFilter</b> , <b>ISpecifyPropertyPages</b> , <b>IMP2Requant</b>
Filter CLSID	CLSID_MP2REQ {74D31C33-5FCF-4b92-A02D-594D56B1B00B}
Input Pin Media Types	Major type: MEDIATYPE_Video Subtype: MEDIASUBTYPE_MPEG2_VIDEO Formattype: FORMAT_MPEG2_VIDEO
Output Pin Media Types	Major type: MEDIATYPE_Video Subtype: MEDIASUBTYPE_MPEG2_VIDEO Formattype: FORMAT_MPEG2_VIDEO
Property Page CLSID	CLSID_MP2REQ_PPage {435B92A9-27C9-4821-8B45-60A467E5DAAB}
Executable	SMM_MP2Req.ax
Merit	MERIT_DO_NOT_USE
Filter Category	CLSID_LegacyAmFilterCategory

### IBaseFilter, ISpecifyPropertyPages interfaces

The *IBaseFilter* and the *ISpecifyPropertyPages* are defined by DirectShow® and COM documentation.

### IMP2Requant interface

The *IMP2Requant* interface is a specific to the SolveigMM MPEG-2 Requantizer Filter. The *IMP2Requant* interface is exposed by the SolveigMM MPEG-2 Requantizer Filter and provides methods to set the filter parameters.

IID of IMP2Requant interface is:

IID\_IMP2Requant  
{8FF0EB9B-8812-4f27-A26F-A6975562EF38}

### Methods in VTable order

IUnknown methods	Description
<b>QueryInterface</b>	Retrieves pointers to supported interfaces
<b>AddRef</b>	Increments reference count
<b>Release</b>	Decrements reference count
IMP2Requant methods	Description
<a href="#">get_CompRatio</a>	Retrieves a current compression ratio set
<a href="#">put_CompRatio</a>	Specifies a compression ratio
<a href="#">get_AdvSettings</a>	Retrieves current advanced settings set
<a href="#">put_AdvSettings</a>	Specifies advanced settings
<a href="#">get_NumPassedFrames</a>	Retrieves a current number of video frames processed by the filter

**IMP2Requant::get\_CompRatio**[IMP2Requant interface](#)

The **get\_CompRatio** method retrieves a current compression ratio set

**Syntax**

```
HRESULT get_CompRatio (  
    int* pnRatio  
);
```

**Parameters**

pnRatio  
[out] – Pointer to a variable that receives compression ratio information

**Return Value**

Returns E\_POINTER if the pointer argument is NULL and S\_OK otherwise.

**Remarks**

For more information about a compression ratio see [put\\_CompRatio](#) remarks.

---

**IMP2Requant::put\_CompRatio**[IMP2Requant interface](#)

The **put\_CompRatio** method specifies a compression ratio value to the filter

**Syntax**

```
HRESULT put_CompRatio (  
    int nRatio  
);
```

**Parameters**

nRatio  
[in] – Variable to contain a compression ratio value that should be set to the filter

**Return Value**

Value	Description
E_INVALIDARG	If compression ration is less than 100
S_OK	Success

**Remarks**

Compression ratio is a percentage value reflecting how the filter has to decrease an output video bitrate with respect to an input one.

For example:

- the value 100 means the filter must not accomplish a recompression
- the value 200 means the filter has to try to decrease bitrate twice

**Note:**

SolveigMM MPEG-2 Requantizer implements the smart and fast recompression technology. It does not involve full MPEG-2 video decoding and encoding processes. Concerning a compression ratio the filter tries to solve two tasks: 1) decrease an output bitrate in concordance with the ratio; 2) try to avoid video quality degradation and block artifacts arising. Since these two tasks are contradictory, an output bitrate can differ from a necessary one (as on a compression ratio).

---

**IMP2Requant::get\_AdvSettings**[IMP2Requant interface](#)

The **get\_AdvSettings** method retrieves a current compression ratio set

**Syntax**

---

```
HRESULT get_AdvSettings(  
int nFrameType, int* pnQuantDiv, int* pnCoefNum  
);
```

**Parameters**

nFrameType  
[in] – Variable to contain information about video frame and macroblock type for which other (pnQuantDiv, pnCoefNum) parameters should be retrieved

pnQuantDiv  
[out] – Pointer to a variable that receives a quantizer divisor value set for specified macroblock type of specified video frame type

pnCoefNum  
[out] – Pointer to a variable that receives number of DCT coefficients to be discarded set for specified macroblock type of specified video frame type

**Return Value**

Returns E\_POINTER if the one of a pointer arguments is NULL and S\_OK otherwise.

**Remarks**

For more information about input/output parameters see [put\\_AdvSettings](#) remarks.

---

**IMP2Requant::put\_AdvSettings**

[IMP2Requant interface](#)

The **put\_AdvSettings** method retrieves a current compression ratio set

**Syntax**

```
HRESULT get_AdvSettings(  
int nFrameType, int nQuantDiv, int nCoefNum  
);
```

**Parameters**

nFrameType  
[in] – Variable to contain information about video frame and macroblock type for which other (nQuantDiv, nCoefNum) parameters should be set

nQuantDiv  
[in] – Variable to contain a quantizer divisor value that should be set for a specified macroblock type of specified video frame type

nCoefNum  
[in] – Variable to contain number of DCT coefficients to be discarded that should be set for a specified macroblock type of specified video frame type

**Return Value**

Returns E\_POINTER if the pointer argument is NULL and S\_OK otherwise.

**Remarks**

All the define names listed below are defined in /SDK/Include/int.h header.

The routine put\_AdvSettings allows a developer to adjust separately requantizing settings for each type of macroblock of each type of video frames (table 1 and table 2). For more information about frames and macroblocks see **ISO/IEC 13818-2** standard.

**Table 1. Macroblock types**

Parameter name	Value	Description
REQUANT_PARAM_BLOCK_INTRA	1	Intra coded macroblock
REQUANT_PARAM_BLOCK_NONINTRA	2	Non-intra coded macroblock

**Table 2. Video frame types**

Parameter name	Value	Description
REQUANT_PARAM_FRAME_I	4	I Frame: intra-coded picture
REQUANT_PARAM_FRAME_P	8	P Frame: predictive-coded picture
REQUANT_PARAM_FRAME_B	16	B Frame: bi-directional predictive-coded picture

Quantizer divisor is a percentage value reflecting how the DCT coefficients should be decreased by means of requantizing for set macroblock and frame types.

For example:

- the value 100 means not to requantize DCT coefficients
- the value 200 means to requantize DCT coefficients to decrease values twice

Number of DCT coefficients to be discarded sets number of latest (high-frequency) non-zero DCT coefficients in luminance 8x8 blocks that should be discarded.

**Sample**

```
// setting parameters for intra blocks of I-frames
int nFrameType = ( REQUANT_PARAM_FRAME_I | REQUANT_PARAM_BLOCK_INTRA );
int nQuantRatio = 110;
int nCoefNum = 0;

HRESULT hr = m_pIMP2Req->put_AdvSettings( nFrameType, nQuantRatio, nCoefNum );
if( FAILED( hr ) ){
    return hr;
}

// setting parameters for non-intra blocks of B-frames
nFrameType = ( REQUANT_PARAM_FRAME_B | REQUANT_PARAM_BLOCK_INTRA );
nQuantRatio = 200;
nCoefNum = 1;

HRESULT hr = m_pIMP2Req->put_AdvSettings( nFrameType, nQuantRatio, nCoefNum );
if( FAILED( hr ) ){
    return hr;
}
```

---

**IMP2Requant::get\_NumPassedFrames**

[IMP2Requant interface](#)

The **get\_NumPassedFrames** method retrieves a current number of video frames processed by the filter and frame rate value to be contained in MPEG-2 Video sequence header.

**Syntax**

---

## SolveigMM MPEG-2 Requantizer SDK

```
HRESULT get_NumPassedFrames (  
int* pnFrames, double *pdFrameRate  
);
```

### Parameters

pnFrames

[out] – Pointer to a variable that receives number of video frames processed

pdFrameRate

[out] – Pointer to a variable that receives video frame rate value in frame per second units.

### Return Value

Returns E\_POINTER if one of pointer arguments is NULL and S\_OK otherwise.