

# **SubtitleCreator Subtitle Filter Programmer's Guide**

Version 0.5

**September 22, 2006**

By Manusse ([manusseATusers.sourceforge.net](mailto:manusseATusers.sourceforge.net))

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Filter characteristics</b>	<b>3</b>
2.1	Filter name	3
2.2	Filter properties	3
2.3	Filter interface	3
2.3.1	Description of each method	4
2.3.1.1	SetSubtitleInfo	4
2.3.1.2	SCmalloc	5
2.3.1.3	GetVersion	5
2.3.1.4	DisplayOsd	5
2.3.1.5	GetVideoProps	5
2.3.1.6	ScreenCopy	5
2.3.2	Example of instantiation with a C# application	5
2.4	Filter connection	5
2.5	Typical filter graph	6
2.6	Known incompatibility	6
<b>3</b>	<b>Conclusion</b>	<b>6</b>

# 1 Introduction

SubtitleCreator (<http://sourceforge.net/projects/subtitlecreator/>) is an open source program that allows to create and manipulate SRT subtitle files to create SUP files that can be used by different authoring softwares such as [IfoEdit](#) or [ReJig](#).

I thought it would be interesting if SubtitleCreator would also allow to preview the subtitles on top of the DVD video.

As subtitlecreator was already displaying the video, I would just have to look at how to overlay some bitmap on top of it.

After some investigations, it appeared that a DirectShow filter could be one solution. As I was interested in learning this technology, and as Erik Vullings, the author of SubtitleCreator was interested I started learning about it.

The result is SC Subtitle filter.

This code is based on the grabber sample as in the book of Mark Pesce « Programming DirectShow and Digital Video » and also for some parts on the DirectShow Filter Wizard Version 7.1.

This documentation is related to version 0.5 of the filter

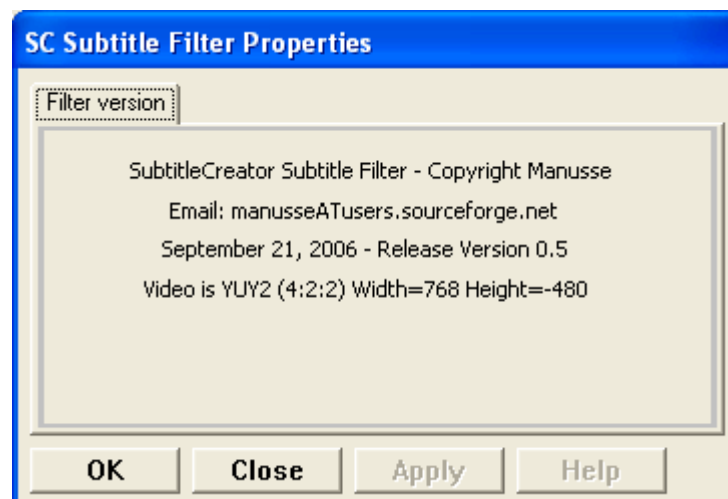
## 2 Filter characteristics

### 2.1 Filter name

The filter file name is **SCSubtitleFilter.ax**

### 2.2 Filter properties

When you instantiate the filter with graphedit, you can display its properties :



The version of the filter is displayed.

If the filter is part of a running graph, it will also display some useful information about the used video buffer.

### 2.3 Filter interface

The filter is written in C++. The interface of the filter consists of different methods and GUIDs :

```
// {98D7BAC8-96D6-427f-A73F-1B7364A59D7D}  
DEFINE_GUID(CLSID_SCSUBTITLEFILTER,
```

```

0x98d7bac8, 0x96d6, 0x427f, 0xa7, 0x3f, 0x1b, 0x73, 0x64, 0xa5, 0x9d,
0x7d);

// {EB4D324F-2CF6-4dbb-A55F-FAB59CB0F7F2}
DEFINE_GUID(CLSID_SCSSubtitleFilterPropertyPage,
0xeb4d324f, 0x2cf6, 0x4dbb, 0xa5, 0x5f, 0xfa, 0xb5, 0x9c, 0xb0, 0xf7,
0xf2);

// {AA11C9FC-FB79-4e28-A33F-69C0C8AEE061}
DEFINE_GUID(IID_SCSSubtitleFilter,
0xaa11c9fc, 0xfb79, 0x4e28, 0xa3, 0x3f, 0x69, 0xc0, 0xc8, 0xae, 0xe0,
0x61);

// We define the interface the app can use to program us
MIDL_INTERFACE("AA11C9FC-FB79-4e28-A33F-69C0C8AEE061")
ISCSSubtitleFilter : public IUnknown
{
    public:
        virtual HRESULT STDMETHODCALLTYPE SetSubtitleInfo(
            int Xpos,
            int Ypos,
            int XImgSize,
            int YImgSize,
            unsigned char *pRGBABuffer) = 0;

    public:
        virtual HRESULT STDMETHODCALLTYPE SCmalloc(
            int Nbytes) = 0;

    public:
        virtual HRESULT STDMETHODCALLTYPE GetVersion(void) = 0;

    public:
        virtual HRESULT STDMETHODCALLTYPE DisplayOsd(
            int ON_OFF) = 0;

    public:
        virtual HRESULT STDMETHODCALLTYPE GetVideoProps(void) = 0;

    public:
        virtual HRESULT STDMETHODCALLTYPE ScreenCopy(
            char *pFileName) = 0;

```

### 2.3.1 Description of each method

#### 2.3.1.1 SetSubtitleInfo

This method is used by the application to send a bitmap to the filter.

Xpos : The horizontal position of the filter relative to the video window

Ypos : The vertical position of the filter relative to the video window

XimgSize : The horizontal dimension of the bitmap in pixels. This dimension MUST be a multiple of 8.

YimgSize : The vertical dimension of the bitmap in pixels This dimension MUST be a multiple of 2.

pRGBABuffer : A byte pointer that addresses the first byte of the bitmap to be displayed. The data are arranged as Blue, Green, Red<sup>1</sup>, Alpha (transparency). If Alpha is 0, then it is a transparent pixel, else whatever the value, it is an opaque one.

Returns 0 if OK.

---

<sup>1</sup> In version 0.4, pixel was expected as Red, Green, Blue, Alpha which was not compatible with C# default mode. Was changed after a request from EV.

### 2.3.1.2 Scalloc

This method is called to allocate some memory in the filter space for transferring the bitmap data. This allows to easily use some unmanaged memory.

Returns as an integer the address of the allocated memory. This buffer is not initialized.

The memory is freed when the filter is unloaded from memory. There is no need to specifically free the memory.

### 2.3.1.3 GetVersion

Returns as an integer the address of a string holding the version information of the filter. (This is an ANSI string, not UNICODE)

Typically : 0.5

### 2.3.1.4 DisplayOsd

Not used for the moment.

### 2.3.1.5 GetVideoProps

Returns as an integer the address of a buffer holding the width, then the height (integers) of the actual used video buffer. This information can be used to stretch the bitmap if needed.

### 2.3.1.6 ScreenCopy

This function will create a bitmap holding a screencopy of the video including the overlaid bitmap. The filename must be passed as a parameter. This methods will produce a bmp file only with the debug version of the filter.

pFileName : A byte pointer that holds an ANSI string with the bmp filename. (Should end by .bmp)

## 2.3.2 Example of instanciation with a C# application

This example comes from SubtitleCreator :

```
[ComImport,
Guid("98D7BAC8-96D6-427f-A73F-1B7364A59D7D")]
public class SCSubtitleFilter
{
    [ComVisible(true), ComImport,
    Guid("A411C9FC-FB79-4e28-A33F-69C0C8AEE061"),
    InterfaceType( ComInterfaceType.InterfaceIsIUnknown )]
    public interface ISCSUBTITLEFILTER
    {
        [PreserveSig]
        int SetSubtitleInfo(int Xpos, int Ypos, int XImgSize, int YImgSize, IntPtr pRGBABuffer);

        [PreserveSig]
        int Scalloc( int Nbytes );

        [PreserveSig]
        int GetVersion();

        [PreserveSig]
        int DisplayOsd( int ON_OFF );

        [PreserveSig]
        int GetVideoProps();

        [PreserveSig]
        int ScreenCopy(IntPtr pFileName);
    }
}
```

## 2.4 Filter connection

The filter has one input and one output. The input accepts a video stream typically coming from a video decoder and mixes it with a bitmap.

This filter can accept these kind of inputs :

**Type :** MEDIATYPE\_Video

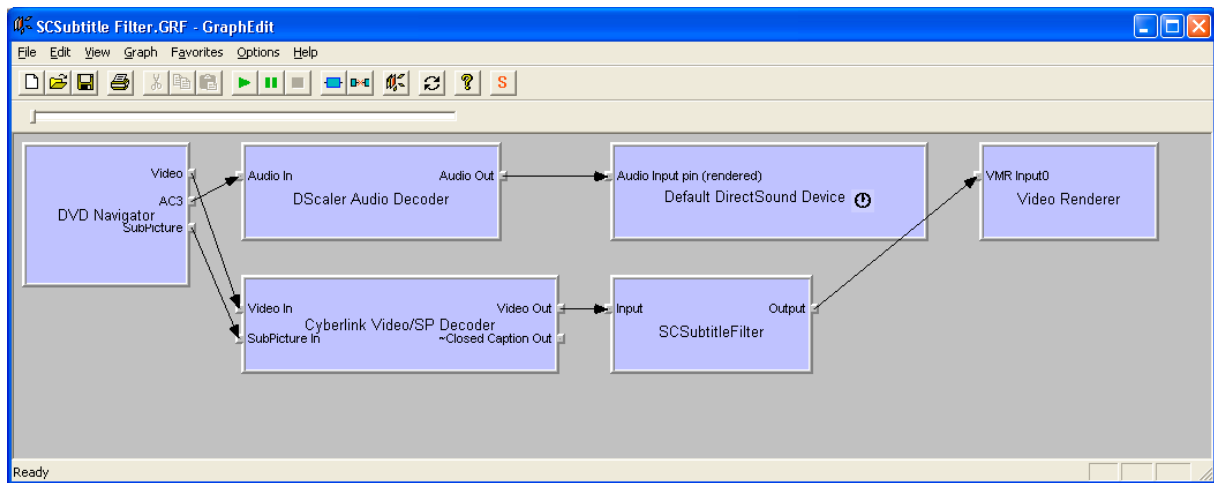
**SubType :** MEDIASUBTYPE\_YUY2 or MEDIASUBTYPE\_YV12

**FormatType** FORMAT\_VideoInfo2

It will refuse connections with inputs of other types. Its output will be of the same type as its input.

## 2.5 Typical filter graph

Below the copy of a typical filter graph as seen in graphedit.



## 2.6 Known incompatibility

This filter can't handle video size reconfiguration. This means that it can't work with all kind of Video Decoders.

The Video Decoders that are supposed to work quite well with this filter are :

- Cyberlink Video/SP Decoder (tested with versions 4 and 6)
- Fraunhofer Video Decoder

Some others may work well also.

The Video Decoders that are (at present) not compatible with the filter :

- DScaler Mpeg2 Video Decoder
- InterVideo Video Decoder (Seem to be compatible for NTSC DVDs)

## 3 Conclusion

Feel free to use this filter in your applications as it is open source freeware. You need my authorization only for use in commercial products. In this case, please contact me at the email address given at the beginning of this document.