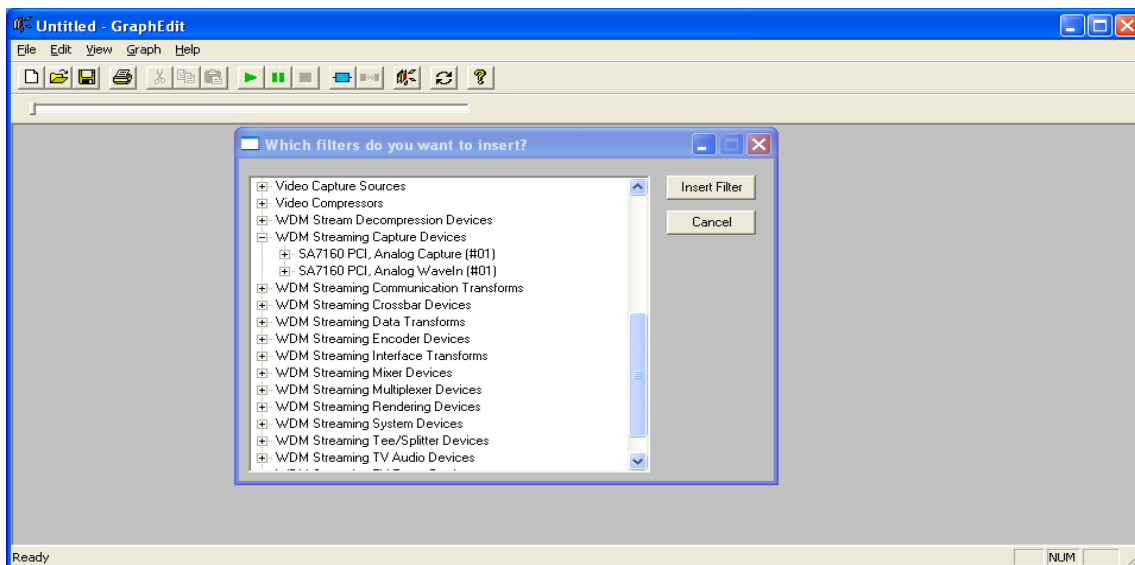


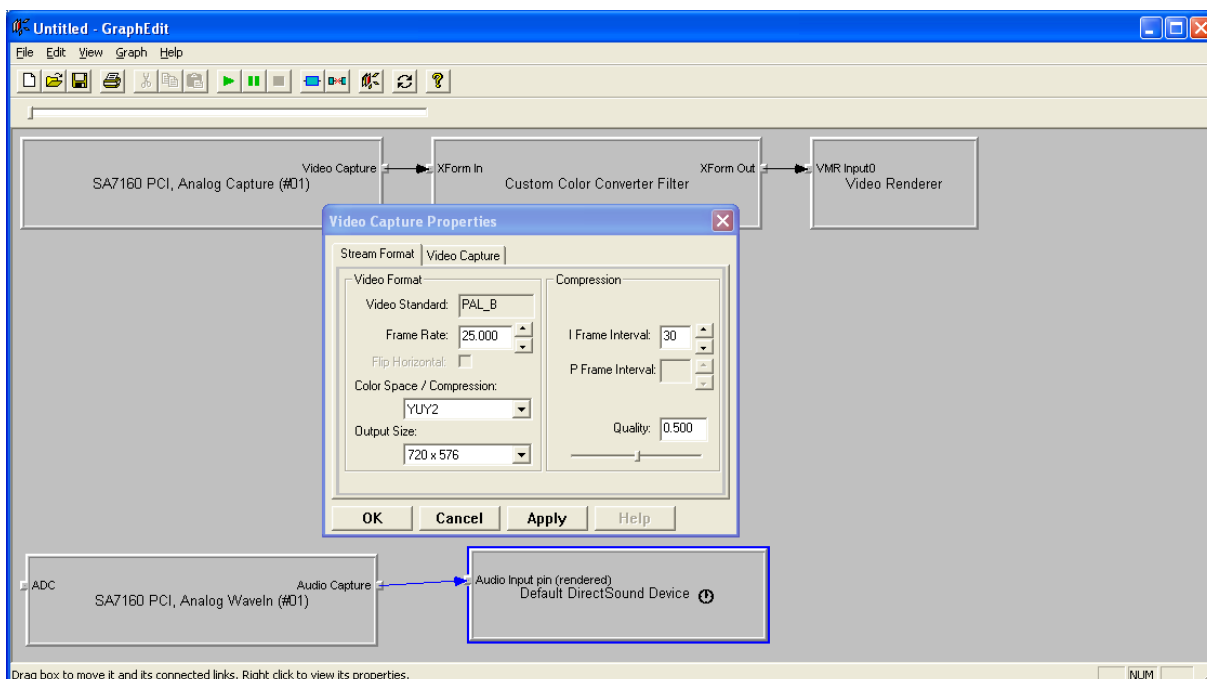
PD652 DirectShow Software Programming Guide

Customer uses DirectShow to develop software can bypass our SDK to access DC1150 directly. Majority of device properties is implemented by Microsoft DirectShow standard interface. Software developer can refer Section 1 and Section 2 to control them. Other custom properties are implemented by IKsPropertySet interface. The interface can be queried from our capture source filter. Section 3 will describe how to access them in detail.

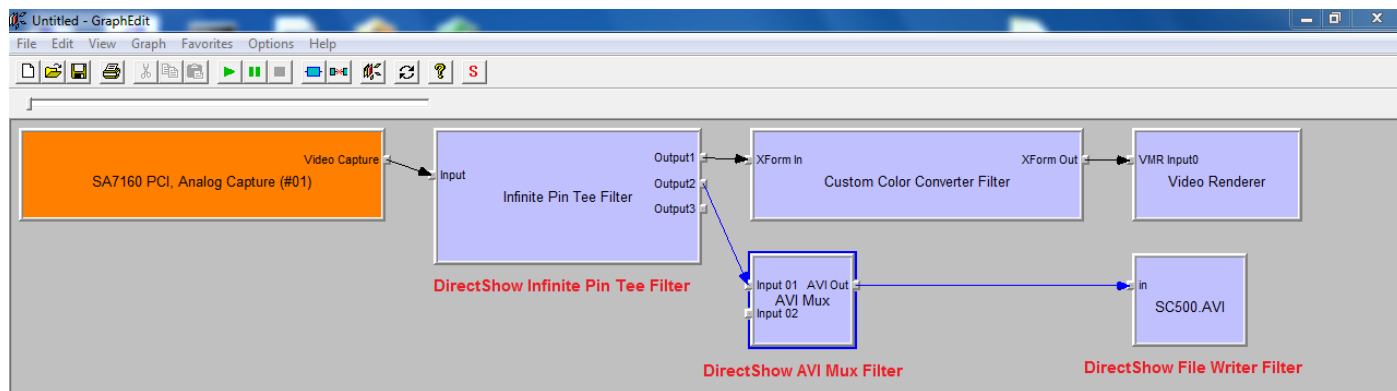
All filter names are "DC1150 USB, Analog Capture (#XX)" for video, and "DC1150 USB, Analog WaveIn (#XX)" for audio. They are registered at "WDM Streaming Captures Devices" category.



Here, the video format is YUY2 and audio format is PCM. The connection of filters are as:



Moreover, customer wants to use graphedit to save raw video stream into AVI can reference as below:



1. ACCESS VIDEO STANDARD (IAMAnalogVideoDecoder)

The video standard is implemented by IAMAnalogVideoDecoder interface. Customer must to setup the correct standard before accessing video format. For example, the 720X480@30fps format is only implemented under NTSC, and the 720x576@25fps format is only implemented under PALB.

EXAMPLE#01: SET STANDARD TO NTSC.

```
m_pCommonCaptureGraphBuilder2->FindInterface( NULL,
                                                NULL,
                                                m_pVideoCaptureSourceBaseFilter,
                                                IID_IAMAnalogVideoDecoder,
                                                (VOID **) (&m_pAMAnalogVideoDecoder) );
m_pAMAnalogVideoDecoder->put_TVFormat( AnalogVideo_NTSC_M );
```

2. ACCESS OUTPUT FORMAT OF CAPTURE PIN (IAMStreamConfig)

To get/set output format of capture pin, customer can use IAMStreamConfig interface.

EXAMPLE#01: SET VIDEO OUTPUT FORAMT TO 720X480 AT 30FPS.

```
m_pCommonCaptureGraphBuilder2->FindInterface( &LOOK_DOWNSTREAM_ONLY,
                                                NULL,
                                                m_pVideoCaptureSourceBaseFilter,
                                                IID_IAMStreamConfig,
                                                (VOID **)( &m_pAMStreamConfig ) );

AM_MEDIA_TYPE * pmt = NULL;
m_pAMStreamConfig->GetFormat( &pmt );
((VIDEOINFOHEADER *) (pmt->pbFormat))->bmiHeader.biCompression = MAKEFOURCC('Y', 'U', 'Y', '2');
((VIDEOINFOHEADER *) (pmt->pbFormat))->bmiHeader.biHeight = 720;
((VIDEOINFOHEADER *) (pmt->pbFormat))->bmiHeader.biWidth = 480;
((VIDEOINFOHEADER *) (pmt->pbFormat))->bmiHeader.biBitCount = 16;
((VIDEOINFOHEADER *) (pmt->pbFormat))->bmiHeader.biSizeImage = 720* 480 * 16 / 8;
((VIDEOINFOHEADER *) (pmt->pbFormat))->AvgTimePerFrame = (ULONG) (INT) (10000000.0 / 30.000);
((VIDEOINFOHEADER *) (pmt->pbFormat))->dwBitRate = (ULONG) (INT) ( 720* 480 * 16 * 30.000);
m_pAMStreamConfig->SetFormat( pmt );
DeleteMediaType( pmt );
```

EXAMPLE#02: SET AUDIO OUTPUT FORAMT TO SETERO, 16BITS, AND 48000HZ.

```
m_pCommonCaptureGraphBuilder2->FindInterface( &LOOK_DOWNSTREAM_ONLY,
                                                NULL,
                                                m_pAudioCaptureSourceBaseFilter,
                                                IID_IAMStreamConfig,
                                                (VOID **)( &m_pAMStreamConfig ) );

AM_MEDIA_TYPE * pmt = NULL;
m_pAMStreamConfig->GetFormat( &pmt );
((WAVEFORMATEX *) (pmt->pbFormat))->nChannels = (USHORT) (2);
((WAVEFORMATEX *) (pmt->pbFormat))->wBitsPerSample = (USHORT) (16);
((WAVEFORMATEX *) (pmt->pbFormat))->nSamplesPerSec = (ULONG) (48000);
((WAVEFORMATEX *) (pmt->pbFormat))->nBlockAlign = (USHORT) (2 * 16 / 8);
((WAVEFORMATEX *) (pmt->pbFormat))->nAvgBytesPerSec = (ULONG) (2 * 16 * 48000 / 8);
m_pAMStreamConfig->SetFormat( pmt );
DeleteMediaType( pmt );
```

3 Customer Property Access

Customer can access all custom properties by IKsPropertySet, the parameter rguidPropSet of IKsPropertySet::Set/Get function, is defined as below:

```
GUID PROPSETID_AMEBDAD_CUSTOM_PROP =  
{ 0xD1E5209F, 0x68FD, 0x4529, 0xBE, 0xE0, 0x5E, 0x7A, 0x1F, 0x47, 0x92, 0x11 };
```

All custom properties are defined as below:

```
typedef enum {  
    KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_DEINTERLACE_TYPE    = 200,  
    KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_INPUT                = 201,  
    KSPROPERTY_CUSTOM_GET_ANALOG_VIDEO_MACROVISION           = 202,  
} KSPROPERTY_AMEBDAD_CUSTOM;
```

3.1. KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_DEINTERLACE_TYPE (200)

The property allows you to get/change deinterlacer. We can support total 4 kinds of deinterlace method, Bob, Weave, Low Motion, and Blending.

SUPPORT VALUE: 0: Bob
 1: Weave (Turn Off)
 2: Motion Adapter
 3: Blending

EXAMPLE#01: TURN OFF DEINTERLACE FUNCTION.

```
ULONG input = 1;
m_pKsPropertySet->Set( PROPSETID_AMEBDAD_CUSTOM_PROP,
                        KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_DEINTERLACE_TYPE,
                        NULL, 0,
                        &input, sizeof(ULONG) );
```

EXAMPLE#02: ENABLE BLENDING ALGORITHM.

```
ULONG input = 3;
m_pKsPropertySet->Set( PROPSETID_AMEBDAD_CUSTOM_PROP,
                        KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_DEINTERLACE_TYPE,
                        NULL, 0,
                        &input, sizeof(ULONG) );
```

EXAMPLE#03: GET CURRENT DEINTERLACE METHOD.

```
ULONG input = 0;
m_pKsPropertySet->Get( PROPSETID_AMEBDAD_CUSTOM_PROP,
                       KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_DEINTERLACE_TYPE,
                       NULL, 0,
                       &input, sizeof(ULONG), &temp );
```

3.2. KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_INPUT (201)

The property allows you to get/change current video input source. We can support total 5 kinds of video input sources, HDMI, DVI-D, Components, DVI-A, and SDI.

SUPPORT VALUE: 0: Composite
 1: S-Video

EXAMPLE#01: SET INPUT TO S-VIDEO.

```
ULONG input = 1;
m_pKsPropertySet->Set( PROPSETID_AMEBDAD_CUSTOM_PROP,
                        KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_INPUT,
                        NULL, 0,
                        &input, sizeof(ULONG) );
```

EXAMPLE#02: CHANGE INPUT TO COMPOSITE.

```
ULONG input = 0;
m_pKsPropertySet->Set( PROPSETID_AMEBDAD_CUSTOM_PROP,
                        KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_INPUT,
                        NULL, 0,
                        &input, sizeof(ULONG) );
```

EXAMPLE#03: GET CURRENT VIDEO INPUT SOURCE.

```
ULONG input = 0;
m_pKsPropertySet->Get( PROPSETID_AMEBDAD_CUSTOM_PROP,
                        KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_INPUT,
                        NULL, 0,
                        &input, sizeof(ULONG), &temp );
```

3.3. KSPROPERTY_CUSTOM_GET_ANALOG_VIDEO_MACROVISION (202) (READ ONLY)

The property allows you to detect if the input's media content owns MarcoVision protection.

Note!! To protect the content license, all behaviors in software porting should be complied with MacroVision rules. Detect in any registered content of MarcoVision, please disable the recording function in software.

SUPPORT VALUE: 0, 1 - NO ~ YES

EXAMPLE#01: GET MACROVISION PROTECT STATUS.

```
ULONG HDCP = 0;
```

[illegible]

4. Application Note for DirectShow Developer

The developer who uses DirectShow to access our capture source filter need check the frame size in the callback function of your SampleGrabber class. If the frame size is 0 bytes, it means the frame is one bad frame. You should drop it. More detail, please check with our engineer team directly.