

# Visual C++编程实现摄像头视频捕捉

## 前言

DirectShow 是微软公司提供的一套在 Windows 平台上进行流媒体处理的开发包，与 DirectX 开发包一起发布。DirectShow 为多媒体流的捕捉和回放提供了强有力的支持。用 DirectShow 开发应用程序，我们可以很方便地从支持 WDM 驱动模型的采集卡上捕获数据，并且进行相应的后期处理乃至存储到文件中。

DirectShow 是基于 COM 的，为了编写 DirectShow 应用程序，需要了解 COM 客户程序编写的基础知识。DirectShow 提供了大量的接口，但在编程中发现还是不够方便，如果能构建一个视频捕捉类把常用的一些动作封装起来，那么就更方便了。

## 编程思路

为了更加容易建立视频捕捉应用程序，DirectShow 提供了一个叫做 Capture Graph Builder 的对象，Capture Graph Builder 提供 ICaptureGraphBuilder2 接口，该接口可以建立和控制 Capture Graph。

建立视频捕捉程序，必须首先获取并初始化 ICaptureGraphBuilder2 接口，然后选择一个适当的视频捕捉设备。选择好设备后，为该设备创建 Capture filter，然后调用 AddFilter 把 Capture filter 添加到 Filter Graph。

如果仅仅希望用摄像头来进行实时监控的话，只需要在上面的基础上调用 ICaptureGraphBuilder2::RenderStream 就可以了：

```
ICaptureGraphBuilder2 *pBuild; // Capture Graph Builder
//省略初始化部分代码
IBaseFilter *pCap; // Video capture filter.
//省略初始化和添加到 Filter Graph 部分代码
pBuild->RenderStream(&PIN_CATEGORY_PREVIEW, &MEDIATYPE_Video, pCap, NULL,
NULL);
```

DirectShow 提供了一个捕捉静态图像的方法：使用 Sample Grabber filter。依次按照以下三个步骤就可以了：

第一步，定义一个类实现 Sample Grabber 的回调接口 ISampleGrabberCB：

```
class CSampleGrabberCB : public ISampleGrabberCB
{
    //在后面提供的类中具体完成
}
CSampleGrabberCB mCB;
```

第二步、调用 RenderStream 依次把 Still pin、Sample Grabber 和系统默认 Renderer Filter 连接起来。

第三步、配置 Sample Grabber 以捕获数据。

视频捕捉类 CCAptureVideo 的具体实现

// CCAptureVideo 视频捕捉类头文件

////////////////////////////////////

#if !defined(AFX\_CAPTUREVIDEO\_H\_\_F5345AA4\_A39F\_4B07\_B843\_3D87C4287AA0\_\_IN  
CLUDED\_)

#define

AFX\_CAPTUREVIDEO\_H\_\_F5345AA4\_A39F\_4B07\_B843\_3D87C4287AA0\_\_INCLUDED\_

////////////////////////////////////

// CaptureVideo.h : header file

////////////////////////////////////

#if \_MSC\_VER > 1000

#pragma once

#endif // \_MSC\_VER > 1000

#include <atlbase.h>

#include <windows.h>

#include <dshow.h>

#ifndef SAFE\_RELEASE

#define SAFE\_RELEASE( x ) \

if ( NULL != x ) \

{ \

    x->Release(); \

    x = NULL; \

}

#endif

class CSampleGrabberCB;

class CCAptureVideo : public CWnd

{

    friend class CSampleGrabberCB;

public:

    void GrabOneFrame(BOOL bGrab);

    HRESULT Init(int iDeviceID,HWND hWnd);

    int EnumDevices(HWND hList);

    CCAptureVideo();

    virtual ~CCAptureVideo();

private:

    HWND m\_hWnd;

    IGraphBuilder \*m\_pGB;

    ICaptureGraphBuilder2\* m\_pCapture;

    IBaseFilter\* m\_pBF;

    IMediaControl\* m\_pMC;

    IVideoWindow\* m\_pVW;

    CComPtr<ISampleGrabber> m\_pGrabber;

protected:

    void FreeMediaType(AM\_MEDIA\_TYPE& mt);

```

    bool BindFilter(int deviceId, IBaseFilter **pFilter);
    void ResizeVideoWindow();
    HRESULT SetupVideoWindow();
    HRESULT InitCaptureGraphBuilder();
};
#endif
// !defined(AFX_CAPTUREVIDEO_H__F5345AA4_A39F_4B07_B843_3D87C4287AA0__INC
LUDED_)
//-----
// CCaptureVideo 视频捕捉类实现文件 CaptureVideo.cpp
//-----
// CaptureVideo.cpp: implementation of the CCaptureVideo class.
//
////////////////////////////////////
#include "stdafx.h"
#include "CaptureVideo.h"
#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
BOOL bOneShot=FALSE;//全局变量
class CSampleGrabberCB : public ISampleGrabberCB
{
public:
    long lWidth;
    long lHeight;
    TCHAR m_szFileName[MAX_PATH];// 位图文件名称
    CSampleGrabberCB(){
        strcpy(m_szFileName, "c:\\donaldo.bmp");
    }
    STDMETHODCALLTYPE AddRef() { return 2; }
    STDMETHODCALLTYPE Release() { return 1; }
    STDMETHODCALLTYPE QueryInterface(REFIID riid, void **ppv){
        if( riid == IID_ISampleGrabberCB || riid == IID_IUnknown ){
            *ppv = (void *) static_cast<ISampleGrabberCB*>( this );
            return NOERROR;
        }
        return E_NOINTERFACE;
    }
    STDMETHODCALLTYPE SampleCB( double SampleTime, IMediaSample * pSample ){
        return 0;
    }
    STDMETHODCALLTYPE BufferCB( double dblSampleTime, BYTE * pBuffer, long lBufferSize ){

```



```

{
//COM Library Intialization
if(FAILED(CoInitialize(NULL))) /*, COINIT_APARTMENTTHREADED))*/
{
    AfxMessageBox("CoInitialize Failed!\r\n");
    return;
}
m_hWnd = NULL;
m_pVW = NULL;
m_pMC = NULL;
m_pGB = NULL;
m_pCapture = NULL;
}
CCaptureVideo::~CCaptureVideo()
{
// Stop media playback
if(m_pMC)m_pMC->Stop();
if(m_pVW){
    m_pVW->put_Visible(OAFALSE);
    m_pVW->put_Owner(NULL);
}
SAFE_RELEASE(m_pCapture);
SAFE_RELEASE(m_pMC);
SAFE_RELEASE(m_pGB);
SAFE_RELEASE(m_pBF);
CoUninitialize();
}
int CCaptureVideo::EnumDevices(HWND hList)
{
    if (!hList)
        return -1;
    int id = 0;

//枚举视频捕捉设备
    ICreateDevEnum *pCreateDevEnum;
    HRESULT hr = CoCreateInstance(CLSID_SystemDeviceEnum, NULL,
CLSCTX_INPROC_SERVER, IID_ICreateDevEnum, (void**)&pCreateDevEnum);

    if (hr != NOERROR)return -1;
    CComPtr<IEnumMoniker> pEm;
    hr = pCreateDevEnum->CreateClassEnumerator(CLSID_VideoInputDeviceCategory,&pEm, 0);

    if (hr != NOERROR)return -1;
    pEm->Reset();
}

```

```

ULONG cFetched;
IMoniker *pM;
while(hr = pEm->Next(1, &pM, &cFetched), hr==S_OK)
{
    IPropertyBag *pBag;
    hr = pM->BindToStorage(0, 0, IID_IPropertyBag, (void **)&pBag);
    if(SUCCEEDED(hr))
    {
        VARIANT var;
        var.vt = VT_BSTR;
        hr = pBag->Read(L"FriendlyName", &var, NULL);
        if (hr == NOERROR)
        {
            TCHAR str[2048];
            id++;
            WideCharToMultiByte(CP_ACP,0,var.bstrVal, -1, str, 2048, NULL, NULL);
            ::SendMessage(hList, CB_ADDSTRING, 0,(LPARAM)str);
            SysFreeString(var.bstrVal);
        }
        pBag->Release();
    }
    pM->Release();
}
return id;
}
HRESULT CCaptureVideo::Init(int iDeviceID, HWND hWnd)
{
    HRESULT hr;
    hr = InitCaptureGraphBuilder();
    if (FAILED(hr)){
        AfxMessageBox("Failed to get video interfaces!");
        return hr;
    }
    // Bind Device Filter. We know the device because the id was passed in
    if(!BindFilter(iDeviceID, &m_pBF))return S_FALSE;
    hr = m_pGB->AddFilter(m_pBF, L"Capture Filter");
    // hr = m_pCapture->RenderStream(&PIN_CATEGORY_PREVIEW, &MEDIATYPE_Video,
    // m_pBF, NULL, NULL);
    // create a sample grabber
    hr = m_pGrabber.CoCreateInstance( CLSID_SampleGrabber );
    if( !m_pGrabber ){
        AfxMessageBox("Fail to create SampleGrabber, maybe qedit.dll is not registered?");
        return hr;
    }
}

```

```

CComQIPtr< IBaseFilter, &IID_IBaseFilter > pGrabBase( m_pGrabber );

//设置视频格式
AM_MEDIA_TYPE mt;
ZeroMemory(&mt, sizeof(AM_MEDIA_TYPE));
mt.majortype = MEDIATYPE_Video;
mt.subtype = MEDIASUBTYPE_RGB24;
hr = m_pGrabber->SetMediaType(&mt);

if( FAILED( hr ) ){
    AfxMessageBox("Fail to set media type!");
    return hr;
}
hr = m_pGB->AddFilter( pGrabBase, L"Grabber" );
if( FAILED( hr ) ){
    AfxMessageBox("Fail to put sample grabber in graph");
    return hr;
}

// try to render preview/capture pin
hr = m_pCapture->RenderStream(&PIN_CATEGORY_PREVIEW,
&MEDIATYPE_Video,m_pBF,pGrabBase,NULL);
if( FAILED( hr ) )
hr = m_pCapture->RenderStream(&PIN_CATEGORY_CAPTURE,
&MEDIATYPE_Video,m_pBF,pGrabBase,NULL);

if( FAILED( hr ) ){
    AfxMessageBox("Can't build the graph");
    return hr;
}

hr = m_pGrabber->GetConnectedMediaType( &mt );
if ( FAILED( hr ) ){
    AfxMessageBox("Fail to read the connected media type");
    return hr;
}

VIDEOINFOHEADER * vih = (VIDEOINFOHEADER*) mt.pbFormat;
mCB.lWidth = vih->bmiHeader.biWidth;
mCB.lHeight = vih->bmiHeader.biHeight;
FreeMediaType(mt);
hr = m_pGrabber->SetBufferSamples( FALSE );
hr = m_pGrabber->SetOneShot( FALSE );
hr = m_pGrabber->SetCallback( &mCB, 1 );

```

```

//设置视频捕捉窗口
m_hWnd = hWnd ;
SetupVideoWindow();
hr = m_pMC->Run();//开始视频捕捉
if(FAILED(hr)){AfxMessageBox("Couldn't run the graph!");return hr;}
return S_OK;
}
bool CCaptureVideo::BindFilter(int deviceId, IBaseFilter **pFilter)
{
    if (deviceId < 0)
        return false;

    // enumerate all video capture devices
    CComPtr<ICreateDevEnum> pCreateDevEnum;
        HRESULT hr = CoCreateInstance(CLSID_SystemDeviceEnum, NULL,
CLSCTX_INPROC_SERVER,
IID_ICreateDevEnum, (void**)&pCreateDevEnum);
    if (hr != NOERROR)
    {
        return false;
    }
    CComPtr<IEnumMoniker> pEm;
    hr = pCreateDevEnum->CreateClassEnumerator(CLSID_VideoInputDeviceCategory,&pEm, 0);
    if (hr != NOERROR)
    {
        return false;
    }
    pEm->Reset();
    ULONG cFetched;
    IMoniker *pM;
    int index = 0;
    while(hr = pEm->Next(1, &pM, &cFetched), hr==S_OK, index <= deviceId)
    {
        IPropertyBag *pBag;
        hr = pM->BindToStorage(0, 0, IID_IPropertyBag, (void **)&pBag);
        if(SUCCEEDED(hr))
        {
            VARIANT var;
            var.vt = VT_BSTR;
            hr = pBag->Read(L"FriendlyName", &var, NULL);
            if (hr == NOERROR)
            {
                if (index == deviceId)

```



```

        {
            pM->BindToObject(0, 0, IID_IBaseFilter, (void**)pFilter);
        }
        SysFreeString(var.bstrVal);
    }
    pBag->Release();
}
pM->Release();
index++;
}
return true;
}

```

HRESULT CCaptureVideo::InitCaptureGraphBuilder()

```

{
    HRESULT hr;

    // 创建 IGraphBuilder 接口
    hr=CoCreateInstance(CLSID_FilterGraph,    NULL,    CLSCTX_INPROC_SERVER,
IID_IGraphBuilder, (void **)&m_pGB);
    // 创建 ICaptureGraphBuilder2 接口
    hr = CoCreateInstance (CLSID_CaptureGraphBuilder2 , NULL, CLSCTX_INPROC,
IID_ICaptureGraphBuilder2, (void **) &m_pCapture);
    if (FAILED(hr))return hr;
    m_pCapture->SetFiltergraph(m_pGB);
    hr = m_pGB->QueryInterface(IID_IMediaControl, (void **)&m_pMC);
    if (FAILED(hr))return hr;
    hr = m_pGB->QueryInterface(IID_IVideoWindow, (LPVOID *) &m_pVW);
    if (FAILED(hr))return hr;
    return hr;
}

```

HRESULT CCaptureVideo::SetupVideoWindow()

```

{
    HRESULT hr;
    hr = m_pVW->put_Owner((OAHWND)m_hWnd);
    if (FAILED(hr))return hr;
    hr = m_pVW->put_WindowStyle(WS_CHILD | WS_CLIPCHILDREN);
    if (FAILED(hr))return hr;
    ResizeVideoWindow();
    hr = m_pVW->put_Visible(OATRUE);
    return hr;
}

```

void CCaptureVideo::ResizeVideoWindow()

```

{

```

```

if (m_pVW){
    //让图像充满整个窗口
    CRect rc;
    ::GetClientRect(m_hWnd,&rc);
    m_pVW->SetWindowPosition(0, 0, rc.right, rc.bottom);
}
}
void CCaptureVideo::GrabOneFrame(BOOL bGrab)
{
    bOneShot = bGrab;
}
void CCaptureVideo::FreeMediaType(AM_MEDIA_TYPE& mt)
{
    if (mt.cbFormat != 0) {
        CoTaskMemFree((PVOID)mt.pbFormat);
        // Strictly unnecessary but tidier
        mt.cbFormat = 0;
        mt.pbFormat = NULL;
    }
    if (mt.pUnk != NULL) {
        mt.pUnk->Release();
        mt.pUnk = NULL;
    }
}
}

```

如何使用视频捕捉类 CCaptureVideo

构建 CCaptureVideo 类以后，使用就方便多了，我们在编程中只需要是要下面三个类成员函数就可以实现用摄像头进行视频捕捉：

①int EnumDevices(HWND hList); //hList 是下拉列表框的句柄，本函数用于枚举当前系统安装的所有视频捕捉设备

②HRESULT Init(int iDeviceID,HWND hWnd);//iDeviceID 是视频捕捉设备序号，hWnd 是视频捕捉窗口的句柄

③void GrabOneFrame(BOOL bGrab);//调用 GrabOneFrame(true)就可以捕获当前的静态图像并保存到硬盘上

具体示例：用 MFC AppWizard(exe)创建一个对话框应用程序，取名为 ds，给对话框添加一个下拉列表框(IDC\_COMBO1)、两个按钮（IDC\_PHOTO、IDC\_HAVEALOOK）和一个 Picture 控件（ID:IDC\_STATIC\_SCREEN，Type: Rectangle，Color:Gray）。

## 1、使用向导添加成员变量

```
CStatic m_staticScreen; // IDC_STATIC_SCREEN
CComboBox m_ListCtrl; // IDC_COMBO1
CCaptureVideo m_cap;
```

2、为 BOOL CDsDlg::OnInitDialog()添加如下代码:

```
// TODO: Add extra initialization here
m_cap.EnumDevices (m_ListCtrl);
m_ListCtrl.SetCurSel (0);
```

3、为确定按钮添加代码如下:

```
void CDsDlg::OnOK()
{
    //只需要四行代码就可以进行视频捕捉了
    UpdateData();
    HWND hWnd = m_staticScreen.GetSafeHwnd() ;
    HRESULT hr = m_cap.Init(m_ListCtrl.GetCurSel (),hWnd);
    GetDlgItem(IDOK)->EnableWindow(FALSE);
}
```

4、如果希望捕捉静态图像,为照相按钮添加如下代码:

```
void CDsDlg::OnPhoto()
{
    m_cap.GrabOneFrame(true);
}
```

运行程序时,选定摄像头后只需要按确定就可以了

关于 VC 的设置:

1. 装上 windows directX9 SDK (c:\DXSDK)

路径: include:

C:\DXSDK\Include

C:\DXSDK\SAMPLES\C++\DIRECTSHOW\BASECLASSES

要放在 VC 目录的前面

lib:

C:\DXSDK\Lib

要放在 VC 目录的前面

2. 解决 'ISampleGrabber' : undeclared identifier 问题

在project ->settings -> link 中object/library modules 中填加Strmiids.lib  
Quartz.lib

```
stdafx.h 中有#include <atlbase.h>
           #include <streams.h>
           #include <qedit.h>
```

其中这三个语句要放在后边。

stdafx.h 原来有这些语句：

```
// stdafx.h : include file for standard system include files,
//          or project specific include files that are used frequently, but
//          are changed infrequently
//
#if !defined(AFX_STDAFX_H__C7286142_D886_4302_8952_2DC956B24E2D__INCLUDED_)
#define AFX_STDAFX_H__C7286142_D886_4302_8952_2DC956B24E2D__INCLUDED_
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#define VC_EXTRALEAN          // Exclude rarely-used stuff from Windows headers
#include <afxwin.h>            // MFC core and standard components
#include <afxext.h>           // MFC extensions
#include <afxdisp.h>         // MFC Automation classes
#include <afxdtctl.h>        // MFC support for Internet Explorer 4 Common Controls
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h>          // MFC support for Windows Common Controls
#endif // _AFX_NO_AFXCMN_SUPPORT
```

上面的那三条要放在这些的后边。

3. 从最上面的网址上把 CaptureVideo.cpp 和 CaptureVideo.h 的语句复制粘贴成两个文件并加入 VC 程序中

本文提供的视频捕捉类 CcaptureVideo 和示例，在 Win2K + DirectX9 SDK + VC6 环境下调试通过。注意：编译时需要 Strmiids.lib Quartz.lib 两个库文件(DirectX9 SDK 自带)。