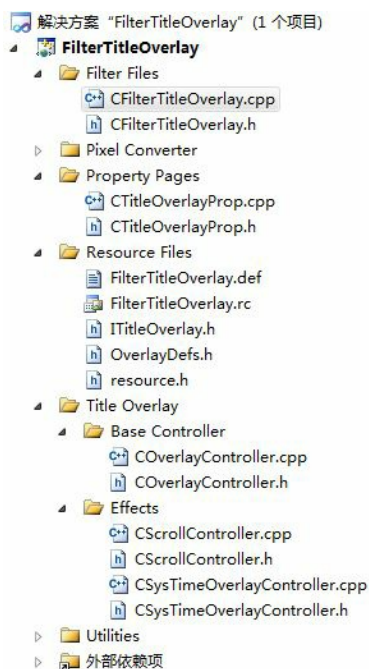


原 DirectShow Filter 开发典型例子分析 ——字幕叠加（FilterTitleOverlay）1

2013年10月09日 12:47:36 阅读数：11017

本文分析一下《DirectShow开发指南》中的一个典型的Transform Filter的例子：字幕叠加(FilterTitleOverlay)。通过分析该例子,我们可以学习到DirectShow Transform Filter 开发的方式。

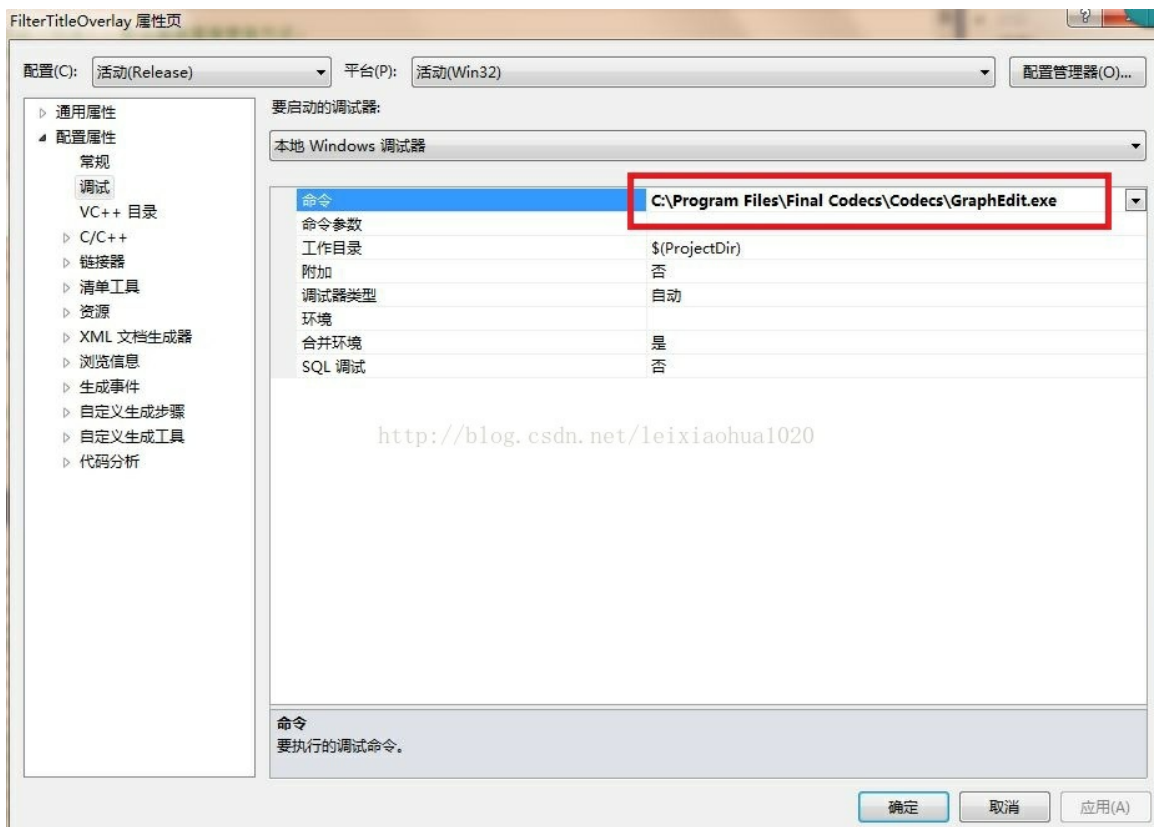
直接打开项目工程（我这里是VC2010），看到项目的结构如下图所示：



先看一下运行的结果：

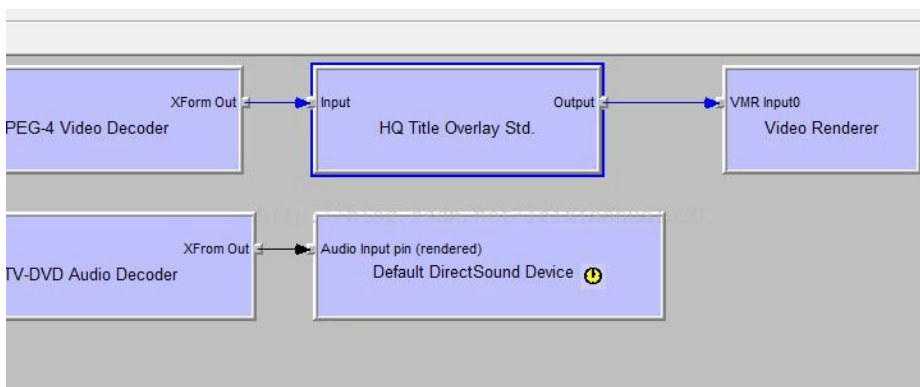
注意，DirectShow的Filter是不可以直接运行进行调试的。一般情况下需要借助于Graphedit.exe这个程序进行调试。当然这不是绝对的，也可以用graph-studio-next这样的开源程序。

选择右键点击工程->属性->调试->命令。在栏中输入Graphedit.exe的路径，如图所示



这样就可以调试Filter了。

拖入一个文件"五月天 咸鱼.mp4"，然后插入本工程的Filter，如图所示。

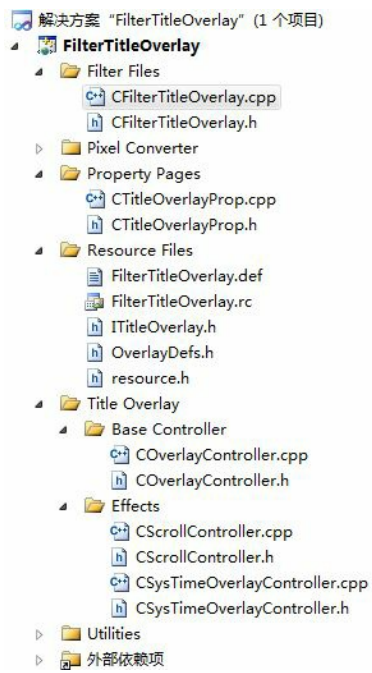


播放视频，效果如图，可见左上角显示出 "Hello, DirectShow!" 的字样。



看完了结果，就要开始分析代码了~

回顾一下工程结构图：



先看一下CFilterTitleOverlay.h（已经在重要的地方加了注释）：

```

1.  //
2.  // CFilterTitleOverlay.h
3.  //
4.
5.  #ifndef __H_CFilterTitleOverlay__
6.  #define __H_CFilterTitleOverlay__
7.
8.  #include "ITitleOverlay.h"
9.  #include "COverlayController.h"
10. #include "OverlayDefs.h"
11.
12. class CFilterTitleOverlay : public CTransInPlaceFilter
13.                          , public ISpecifyPropertyPages
14.                          , public ITitleOverlay
15. {
16. private:
17.     OVERLAY_TYPE      mOverlayType;
18.     COverlayController * mOverlayController;
19.
20.     CCritSec          mITitleOverlaySync;
21.     BOOL               mNeedEstimateFrameRate;
22.
23. private:
24.     CFilterTitleOverlay(TCHAR *tszName, LPUNKNOWN punk, HRESULT *phr);
25.     ~CFilterTitleOverlay();
26.
27.     HRESULT SetInputVideoInfoToController(void);
28.     void ReleaseOverlayController(void);
29.     void SideEffectOverlayTypeChanged(void);
30.
31. public:
32.     static CUnknown * WINAPI CreateInstance(LPUNKNOWN punk, HRESULT *phr);
33.     //说明必须重写NonDelegatingQueryInterface
34.     DECLARE_IUNKNOWN;
35.     // Basic COM - used here to reveal our own interfaces
36.     //暴露接口, 使外部程序可以QueryInterface, 关键!
37.     STDMETHODIMP NonDelegatingQueryInterface(REFIID riid, void **ppv);
38.
39.     // check if you can support mtIn
40.     virtual HRESULT CheckInputType(const CMediaType* mtIn); // PURE
41.     //必须重写的核心函数
42.     virtual HRESULT Transform(IMediaSample *pSample); // PURE
43.
44.     // Delegating methods
45.     virtual HRESULT CompleteConnect(PIN_DIRECTION direction, IPin *pReceivePin);
46.     virtual HRESULT StartStreaming();
47.     virtual HRESULT StopStreaming();
48.
49.     // --- ISpecifyPropertyPages ---
50.     STDMETHODIMP GetPages(CAUUID *pPages);
51.
52.     // --- ITitleOverlay methods ---
53.     //都是接口函数
54.     STDMETHODIMP put_TitleOverlayType(long inOverlayType);
55.     STDMETHODIMP get_TitleOverlayType(long * outOverlayType);
56.     STDMETHODIMP put_TitleOverlayStyle(int inUsingCover);
57.     STDMETHODIMP get_TitleOverlayStyle(int * outUsingCover);
58.     STDMETHODIMP put_Title(const char * inTitle, int inLength);
59.     STDMETHODIMP get_Title(char * outBuffer, int * outLength);
60.     STDMETHODIMP put_TitleColor(BYTE inR, BYTE inG, BYTE inB);
61.     STDMETHODIMP get_TitleColor(BYTE * outR, BYTE * outG, BYTE * outB);
62.     STDMETHODIMP put_TitleStartPosition(POINT inStartPos);
63.     STDMETHODIMP get_TitleStartPosition(POINT * outStartPos);
64.     STDMETHODIMP put_TitleFont(LOGFONT inFont);
65.     STDMETHODIMP get_TitleFont(LOGFONT * outFont);
66.     STDMETHODIMP put_TitleDuration(double inStart, double inEnd);
67.     STDMETHODIMP get_TitleDuration(double * outStart, double * outEnd);
68. };
69.
70. #endif // __H_CFilterTitleOverlay__

```

CFilterTitleOverlay继承了CTransInPlaceFilter, 意味着Transform()函数输入和输出的数据位于同一块内存中。
 以下几个函数是必须有的：

CreateInstance()：创建Filter

NonDelegatingQueryInterface()：暴露接口, 使外部程序可以QueryInterface

CheckInputType()：检查输入类型

Transform()：核心处理函数（字幕叠加）

另外还包含了ITitleOverlay中的函数put_TitleOverlayType()等等一大堆。

下面看一下CFilterTitleOverlay.cpp吧, 先列出注册信息部分：

```

1. //唯一标识符
2. // {E3FB4BFE-8E5C-4aec-8162-7DA55BE486A1}
3. DEFINE_GUID(CLSID_HQTitleOverlay,
4. 0xe3fb4bfe, 0x8e5c, 0x4aec, 0x81, 0x62, 0x7d, 0xa5, 0x5b, 0xe4, 0x86, 0xa1);
5.
6. // {E70FE57A-19AA-4a4c-B39A-408D49D73851}
7. DEFINE_GUID(CLSID_HQTitleOverlayProp,
8. 0xe70fe57a, 0x19aa, 0x4a4c, 0xb3, 0x9a, 0x40, 0x8d, 0x49, 0xd7, 0x38, 0x51);
9.
10.
11. //
12. // setup data
13. //
14. //注册时候的信息
15. const AMOVIESETUP_MEDIATYPE sudPinTypes =
16. {
17.     &MEDIATYPE_NULL,          // Major type
18.     &MEDIASUBTYPE_NULL        // Minor type
19. };
20. //注册时候的信息
21. const AMOVIESETUP_PIN psudPins[] =
22. {
23.     {
24.         L"Input",             // String pin name
25.         FALSE,                // Is it rendered
26.         FALSE,                // Is it an output
27.         FALSE,                // Allowed none
28.         FALSE,                // Allowed many
29.         &CLSID_NULL,          // Connects to filter
30.         L"Output",            // Connects to pin
31.         1,                    // Number of types
32.         &sudPinTypes },        // The pin details
33.     { L"Output",              // String pin name
34.         FALSE,                // Is it rendered
35.         TRUE,                 // Is it an output
36.         FALSE,                // Allowed none
37.         FALSE,                // Allowed many
38.         &CLSID_NULL,          // Connects to filter
39.         L"Input",             // Connects to pin
40.         1,                    // Number of types
41.         &sudPinTypes          // The pin details
42.     }
43. };
44.
45. //注册时候的信息
46. const AMOVIESETUP_FILTER sudFilter =
47. {
48.     &CLSID_HQTitleOverlay,     // Filter CLSID
49.     L"HQ Title Overlay Std.",   // Filter name
50.     MERIT_DO_NOT_USE,          // Its merit
51.     2,                          // Number of pins
52.     psudPins                    // Pin details
53. };
54.
55.
56. // List of class IDs and creator functions for the class factory. This
57. // provides the link between the OLE entry point in the DLL and an object
58. // being created. The class factory will call the static CreateInstance
59. //注意g_Templates名称是固定的
60. CFactoryTemplate g_Templates[] =
61. {
62.     {
63.         L"HQ Title Overlay Std.",
64.         &CLSID_HQTitleOverlay,
65.         CFilterTitleOverlay::CreateInstance,
66.         NULL,
67.         &sudFilter
68.     },
69.     {
70.         L"HQ Title Overlay Property Page",
71.         &CLSID_HQTitleOverlayProp,
72.         CTitleOverlayProp::CreateInstance
73.     }
74. };
75. int g_cTemplates = sizeof(g_Templates) / sizeof(g_Templates[0]);

```

这一部分并不属于CFilterTitleOverlay这个类。主要是DirectShow Filter的一些注册信息。其结构是非常固定的。

再来看看CFilterTitleOverlay中函数实现部分（只列了几个函数，不然内容太多= =）：

CreateInstance()：

```

1. //
2. // CreateInstance
3. //
4. // Override CClassFactory method.
5. // Provide the way for COM to create a CNullInPlace object
6. //
7. //创建
8. CUnknown * WINAPI CFilterTitleOverlay::CreateInstance(LPUNKNOWN punk, HRESULT *phr)
9. {
10. #if 1
11.     //防伪??!!
12.     char szCreatorPath[256], szCreatorName[256];
13.     ::strcpy(szCreatorPath, "");
14.     ::strcpy(szCreatorName, "");
15.     HMODULE hModule = ::GetModuleHandle(NULL);
16.     ::GetModuleFileName(hModule, szCreatorPath, 256);
17.     char * backSlash = ::strrchr(szCreatorPath, '\\');
18.     if (backSlash)
19.     {
20.         strcpy(szCreatorName, backSlash);
21.     }
22.     ::_strlwr(szCreatorName);
23.     // Please specify your app name with lowercase
24.     // 检查调用该Filter的程序
25.     // 一开始调试不了,就卡在这了 ==
26.     if (::strstr(szCreatorName, "graphedit") == NULL &&
27.         ::strstr(szCreatorName, "ourapp") == NULL)
28.     {
29.         *phr = E_FAIL;
30.         return NULL;
31.     }
32. #endif
33.     //通过New对象的方法
34.     CFilterTitleOverlay *pNewObject = new CFilterTitleOverlay(NAME("TitleOverlay"), punk, phr);
35.     if (pNewObject == NULL)
36.     {
37.         *phr = E_OUTOFMEMORY;
38.     }
39.     return pNewObject;
40. }

```

NonDelegatingQueryInterface() :

```

1. //
2. // Basic COM - used here to reveal our own interfaces
3. // 暴露接口,使外部程序可以QueryInterface,关键!
4. STDMETHODIMP CFilterTitleOverlay::NonDelegatingQueryInterface(REFIID riid, void **ppv)
5. {
6.     CheckPointer(ppv, E_POINTER);
7.     //根据不同的REFIID,获得不同的接口指针
8.     if (riid == IID_ISpecifyPropertyPages)
9.     {
10.         return GetInterface((ISpecifyPropertyPages *) this, ppv);
11.     }
12.     else if (riid == IID_ITitleOverlay)
13.     {
14.         return GetInterface((ITitleOverlay *) this, ppv);
15.     }
16.     else
17.     {
18.         //不是以上的REFIID的话,调用父类的
19.         return CTransInPlaceFilter::NonDelegatingQueryInterface(riid, ppv);
20.     }
21. } // NonDelegatingQueryInterface

```



CheckInputType() :

```


1. // Only RGB 32/24/565/555 supported
2. HRESULT CFilterTitleOverlay::CheckInputType(const CMediaType* mtIn)
3. {
4.     // Dynamic format change will never be allowed!
5.     if (IsStopped() && *mtIn->Type() == MEDIATYPE_Video)
6.     {
7.         if (*mtIn->Subtype() == MEDIASUBTYPE_RGB32 ||
8.             *mtIn->Subtype() == MEDIASUBTYPE_RGB24 ||
9.             *mtIn->Subtype() == MEDIASUBTYPE_RGB555 ||
10.            *mtIn->Subtype() == MEDIASUBTYPE_RGB565)
11.         {
12.             return NOERROR;
13.         }
14.     }
15.     return E_INVALIDARG;
16. }

```

Transform() :

```
[cpp]  
1. HRESULT CFilterTitleOverlay::Transform(IMediaSample *pSample)
2. {
3.     // If we can't read frame rate info from input pin's connection media type,
4.     // We estimate it from the first sample's time stamp!
5.     if (mNeedEstimateFrameRate)
6.     {
7.         mNeedEstimateFrameRate = FALSE;
8.         REFERENCE_TIME startTime = 0;
9.         REFERENCE_TIME endTime = 0;
10.        double estimated = 25;
11.        if (SUCCEEDED(pSample->GetTime(&startTime, &endTime)))
12.        {
13.            estimated = 1.0 * UNITS / (endTime - startTime);
14.        }
15.        mOverlayController->SetEstimatedFrameRate(estimated);
16.    }
17.
18.    if (mOverlayType != OT_NONE)
19.    {
20.        //PBYTE是unsigned char
21.        PBYTE pData = NULL;
22.        //获取IMediaSample中的数据
23.        pSample->GetPointer(&pData);
24.        //叠加
25.        mOverlayController->DoTitleOverlay(pData);
26.    }
27.
28.    return NOERROR;
29. }
```

下面列出实现ITitleOverlay接口的函数的实现，就列了一个。

```
[cpp]  
1. STDMETHODCALLTYPE CFilterTitleOverlay::get_Title(char * outBuffer, int * outLength)
2. {
3.     CAutoLock lockit(&mITitleOverlaySync);
4.     *outLength = mOverlayController->GetTitle(outBuffer);
5.     return NOERROR;
6. }
```

暂且分析到这里。

书上提供的代码有误，这是经过修改后，添加了注释的代码：

<http://download.csdn.net/detail/leixiaohua1020/6371819>

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/12498975>

文章标签：[directshow](#) [filter](#) [源代码](#) [分析](#) [字幕](#)

个人分类：[DirectShow](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushide@163.com