

## 原 最简单的基于DirectShow的示例：视频播放器自定义版

2015年01月11日 18:05:19 阅读数：10752

=====

最简单的基于DirectShow的示例文章列表：

[最简单的基于DirectShow的示例：视频播放器](#)

[最简单的基于DirectShow的示例：视频播放器图形界面版](#)

[最简单的基于DirectShow的示例：视频播放器自定义版](#)

[最简单的基于DirectShow的示例：获取Filter信息](#)

=====

本文记录一个简单的基于DirectShow的自定义的视频播放器。这里所说的“自定义播放器”，实际上指的是自己在Filter Graph中手动逐个添加Filter，并且连接这些Filter的后运行的播放器。这么做相对于使用RenderFile()这种“智能”创建Filter Graph的方法来说要复杂不少，但是可以让我们更加了解DirectShow的体系。

□

## 流程图

最简单的基于DirectShow的自定义的视频播放器的流程如下图所示。

□

该流程图中包含如下变量：

- IGraphBuilder \*pGraph：继承自IFilterGraph，用于构建Filter Graph。
- IMediaControl \*pControl：提供和播放控制有关的一些接口。
- IMediaEvent \*pEvent：用来处理Filter Graph发出的事件。
- IBaseFilter \*pF\_source：源Filter。
- IFileSourceFilter\* pFileSource：源Filter的暴露的接口，用于设置输入文件的路径。
- IBaseFilter \*pF\_demuxer：解复用Filter。
- IBaseFilter \*pF\_decoder：解码Filter。
- IBaseFilter \*pF\_render：渲染Filter。
- IPin \*pOut：输出Pin。
- IPin \*pIn：输入Pin。
- IPin \*\*pPin：内部变量Pin。

该流程图大体上可以分成以下步骤：

### (1) 初始化DirectShow

包括以下几个步骤：

- a) CoInitialize()：初始化COM运行环境。
- b) CoCreateInstance(...,pGraph)：用指定的类标识符创建一个Com对象。在这里创建IGraphBuilder。
- c) pGraph->QueryInterface(...,pControl)：通过QueryInterface()查询某个组件是否支持某个特定的接口。在这里查询IMediaControl接口。
- d) pGraph->QueryInterface(...,pEvent)：同上。在这里查询IMediaEvent接口。

### (2) 添加Source Filter

包括以下几个步骤：

- a) CoCreateInstance(...,pF\_source)：创建Source Filter。
- b) pGraph->AddFilter(pF\_source,...)：将Source Filter加入Filter Graph。
- c) pF\_source->QueryInterface(...,pFileSource)：查找Source Filter的IFileSourceFilter接口。
- d) pFileSource->Load(L"xxx.mpg",pF\_source)：调用IFileSourceFilter的Load()方法加载视频文件。

### (3) 添加Demuxer Filter

包括以下几个步骤：

- a) CoCreateInstance(...,pF\_demuxer)：创建Demuxer Filter。
- b) pGraph->AddFilter(pF\_demuxer,...)：将Demuxer Filter加入Filter Graph。

### (4) 添加Decoder Filter

包括以下几个步骤：

- a) CoCreateInstance(...,pF\_decoder)：创建Decoder Filter。
- b) pGraph->AddFilter(pF\_decoder,...)：将Decoder Filter加入Filter Graph。

### (5) 添加Render Filter

包括以下几个步骤：

- a) CoCreateInstance(...,pF\_render)：创建Render Filter。
- b) pGraph->AddFilter(pF\_render,...)：将Render Filter加入Filter Graph。

#### (6) 连接Source Filter和Demuxer Filter

调用了一个函数connect\_filters()用于连接2个Filter。

connect\_filters()的执行步骤如下：

- a) 调用get\_unconnected\_pin()从源Filter中选择一个没有链接的输出Pin。
- b) 调用get\_unconnected\_pin()从目的Filter中选择一个没有链接的输入Pin。
- c) 连接这两个Pin

get\_unconnected\_pin()的执行步骤如下：

- a) 枚举Filter上的Pin。
- b) 遍历这些Pin，查找符合输出方向（通过IPin的QueryDirection()方法），而且没有在使用的Pin（通过IPin的ConnectedTo()方法）。

#### (7) 连接Demuxer Filter和Decoder Filter

过程同上。

#### (8) 连接Decoder Filter和Render Filter

过程同上。

#### (9) 开始播放



包括以下步骤：

- pControl->Run()：开始运行Filter Graph中的所有Filter。
- pEvent->WaitForCompletion()：等待Filter Graph处理完所有数据。

上述步骤可以理解为在GraphEdit软件中分别按照步骤添加以下控件。其中（1）、（2）、（3）、（4）为先添加的4个Filter，（5）、（6）、（7）为Filter之间的连接线。

□

## 源代码

```
[cpp]    
1.  /**  
2.   * 最简单的基于DirectShow的视频播放器 (Custom)  
3.   * Simplest DirectShow Player (Custom)  
4.   *  
5.   * 雷霄骅 Lei Xiaohua  
6.   * leixiaohua1020@126.com  
7.   * 中国传媒大学/数字电视技术  
8.   * Communication University of China / Digital TV Technology  
9.   * http://blog.csdn.net/leixiaohua1020  
10.  *  
11.  * 本程序是一个简单的基于DirectShow的视频播放器。该播放器通过逐个添加  
12.  * 滤镜并连接这些滤镜实现了视频的播放。适合初学者学习DirectShow。  
13.  *  
14.  * This software is a simple video player based on DirectShow.  
15.  * It Add DirectShow Filter Manually and Link the Pins of these filters  
16.  * to play videos.Suitable for the beginner of DirectShow.  
17.  */  
18.  
19.  #include "stdafx.h"  
20.  #include <dshow.h>  
21.  //'1':Add filters manually  
22.  //'0':Add filters automatically  
23.  #define ADD_MANUAL 1  
24.  
25.  //Find unconnect pins  
26.  HRESULT get_unconnected_pin(  
27.      IBaseFilter *pFilter, // Pointer to the filter.  
28.      PIN_DIRECTION PinDir, // Direction of the pin to find.  
29.      IPin **ppPin) // Receives a pointer to the pin.  
30.  {  
31.      *ppPin = 0;  
32.      IEnumPins *pEnum = 0;  
33.      IPin *pPin = 0;  
34.      HRESULT hr = pFilter->EnumPins(&pEnum);  
35.      if (FAILED(hr))  
36.      {  
37.          return hr;  
38.      }  
39.      while (pEnum->Next(1, &pPin, NULL) == S_OK)  
40.      {  
41.          PIN_DIRECTION ThisPinDir;  
42.          pPin->QueryDirection(&ThisPinDir);  
43.          if (ThisPinDir == PinDir)  
44.          {  
45.              IPin *pTmp = 0;  
46.              hr = pPin->ConnectedTo(&pTmp);  
47.              if (SUCCEEDED(hr)) // Already connected, not the pin we want.  
48.              {  
49.                  pTmp->Release();  
50.              }  
51.              else // Unconnected, the pin we want.  
52.              {
```

```

53.         pEnum->Release();
54.         *ppPin = pPin;
55.         return S_OK;
56.     }
57. }
58. pPin->Release();
59. }
60. pEnum->Release();
61. // Did not find a matching pin.
62. return E_FAIL;
63. }
64.
65. //Connect 2 filters
66. HRESULT connect_filters(
67.     IGraphBuilder *pGraph,
68.     IBaseFilter *pSrc,
69.     IBaseFilter *pDest)
70. {
71.     if ((pGraph == NULL) || (pSrc == NULL) || (pDest == NULL))
72.     {
73.         return E_POINTER;
74.     }
75.     //Find Output pin in source filter
76.     IPin *pOut = 0;
77.     HRESULT hr = NULL;
78.     hr=get_unconnected_pin(pSrc, PINDIR_OUTPUT, &pOut);
79.     if (FAILED(hr)){
80.         return hr;
81.     }
82.     //Find Input pin in destination filter
83.     IPin *pIn = 0;
84.     hr = get_unconnected_pin(pDest, PINDIR_INPUT, &pIn);
85.     if (FAILED(hr)){
86.         return hr;
87.     }
88.     //Connect them
89.     hr = pGraph->Connect(pOut, pIn);
90.     pIn->Release();
91.     pOut->Release();
92.     return hr;
93. }
94.
95. int _tmain(int argc, _TCHAR* argv[])
96. {
97.     IGraphBuilder *pGraph = NULL;
98.     IMediaControl *pControl = NULL;
99.     IMediaEvent *pEvent = NULL;
100.    // Init COM
101.    HRESULT hr = CoInitialize(NULL);
102.    if (FAILED(hr)){
103.        printf("Error - Can't init COM.");
104.        return -1;
105.    }
106.
107.    // Create FilterGraph
108.    hr=CoCreateInstance(CLSID_FilterGraph, NULL,CLSCTX_INPROC_SERVER,IID_IGraphBuilder, (void **)&pGraph);
109.    if (FAILED(hr)){
110.        printf("Error - Can't create Filter Graph.");
111.        return -1;
112.    }
113.    // Query Interface
114.    hr = pGraph->QueryInterface(IID_IMediaControl, (void **)&pControl);
115.    hr = pGraph->QueryInterface(IID_IMediaEvent, (void **)&pEvent);
116.
117.    //1. Add Filters=====
118.    //Source
119.    IBaseFilter *pF_source = 0;
120.    hr = CoCreateInstance(CLSID_AsyncReader, 0, CLSCTX_INPROC_SERVER,IID_IBaseFilter, (void **)&pF_source));
121.    if (FAILED(hr)){
122.        printf("Failed to create File Source.\n");
123.        return -1;
124.    }
125.    hr = pGraph->AddFilter(pF_source, L"Lei's Source");
126.    if (FAILED(hr)){
127.        printf("Failed to add File Source to Filter Graph.\n");
128.        return -1;
129.    }
130.    IFileSourceFilter* pFileSource;
131.    pF_source->QueryInterface(IID_IFileSourceFilter, (void **)&pFileSource);
132.    pFileSource->Load(L"cuc_ieschool.mpg", NULL);
133.    pFileSource->Release();
134.
135.    #if ADD_MANUAL
136.    //Demuxer
137.    IBaseFilter *pF_demuxer = 0;
138.    hr = CoCreateInstance(CLSID_MPEG1Splitter, 0, CLSCTX_INPROC_SERVER,IID_IBaseFilter, (void **)&pF_demuxer));
139.    if (FAILED(hr)){
140.        printf("Failed to create Demuxer.\n");
141.        return -1;
142.    }
143.    hr = pGraph->AddFilter(pF_demuxer, L"Lei's Demuxer");

```

```

144.     if (FAILED(hr)){
145.         printf("Failed to add Demuxer to Filter Graph.\n");
146.         return -1;
147.     }
148.     //Decoder
149.     IBaseFilter *pF_decoder = 0;
150.     hr = CoCreateInstance(CLSID_CMpegVideoCodec, 0, CLSCTX_INPROC_SERVER, IID_IBaseFilter, (void**)&pF_decoder));
151.     if (FAILED(hr)){
152.         printf("Failed to create Decoder.\n");
153.         return -1;
154.     }
155.     hr = pGraph->AddFilter(pF_decoder, L"Lei's Decoder");
156.     if (FAILED(hr)){
157.         printf("Failed to add Decoder to Filter Graph.\n");
158.         return -1;
159.     }
160.     //Render
161.     IBaseFilter *pF_render = 0;
162.     hr = CoCreateInstance(CLSID_VideoRenderer, 0, CLSCTX_INPROC_SERVER, IID_IBaseFilter, (void**)&pF_render));
163.     if (FAILED(hr)){
164.         printf("Failed to create Video Render.\n");
165.         return -1;
166.     }
167.     hr = pGraph->AddFilter(pF_render, L"Lei's Render");
168.     if (FAILED(hr)){
169.         printf("Failed to add Video Render to Filter Graph.\n");
170.         return -1;
171.     }
172.     //2. Connect Filters=====
173.     hr = connect_filters(pGraph, pF_source, pF_demuxer);
174.     if (FAILED(hr)){
175.         printf("Failed to link Source and Demuxer.\n");
176.         return -1;
177.     }
178.     hr = connect_filters(pGraph, pF_demuxer, pF_decoder);
179.     if (FAILED(hr)){
180.         printf("Failed to link Demuxer and Decoder.\n");
181.         return -1;
182.     }
183.     hr = connect_filters(pGraph, pF_decoder, pF_render);
184.     if (FAILED(hr)){
185.         printf("Failed to link Decoder and Render.\n");
186.         return -1;
187.     }
188.
189.     pF_source->Release();
190.     pF_demuxer->Release();
191.     pF_decoder->Release();
192.     pF_render->Release();
193. #else
194.     IPin*    Pin;
195.     ULONG    fetched;
196.     // get output pin
197.     IEnumPins* pEnumPins;
198.     hr = pF_source->EnumPins(&pEnumPins);
199.     hr = pEnumPins->Reset();
200.     hr = pEnumPins->Next(1, &Pin, &fetched);
201.     pEnumPins->Release();
202.     // render pin, graph builder automatically complete rest works
203.     hr = pGraph->Render(Pin);
204. #endif
205.
206.     if (SUCCEEDED(hr)){
207.         // Run
208.         hr = pControl->Run();
209.         if (SUCCEEDED(hr)){
210.             long evCode=0;
211.             pEvent->WaitForCompletion(INFINITE, &evCode);
212.         }
213.     }
214.     //Release
215.     pControl->Release();
216.     pEvent->Release();
217.     pGraph->Release();
218.     CoUninitialize();
219.     return 0;
220. }

```

## 运行结果

程序的运行结果如下图所示。运行后会播放“cuc\_ieschool.mpg”文件。需要注意的是，本程序并没有加入音频解码和播放的Filter，所以播放视频的时候是没有声音的。

□

除了手动一个一个添加Filter之外，也可以在获得“源”Filter的Pin之后，直接调用IFilterGraph的Render()方法“智能”自动构建Filter Graph。注意Render()方法和RenderFile()方法是不一样的。RenderFile()是指定一个文件路径后，自动构建整个Filter Graph，相对来说更加简单些；而Render()方法则是首先要创建一个Source Filter之后，

才可以自动构建整个Filter Graph。  
可以通过修改源文件首部的宏定义ADD\_MANUAL来设定是否手动添加Filter，如下所示。

[cpp] [icon] [icon]

```
1.  // '1': Add filters manually
2.  // '0': Add filters automatically
3.  #define ADD_MANUAL 1
```

## 下载

### Simplest DirectShow Example

#### 项目主页

SourceForge：<https://sourceforge.net/projects/simplestdirectshowexample/>

Github：[https://github.com/leixiaohua1020/simplest\\_directshow\\_example](https://github.com/leixiaohua1020/simplest_directshow_example)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_directshow\\_example](http://git.oschina.net/leixiaohua1020/simplest_directshow_example)

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8348163>

本程序包含了DirectShow开发的示例程序。适合DirectShow初学者进行学习。  
它包含了以下几个子程序：  
simplest\_directshow\_player: 最简单的基于DirectShow的视频播放器。  
**simplest\_directshow\_player\_custom: 最简单的基于DirectShow的视频播放器（Custom）。**  
playerGUI: 最简单的基于DirectShow的播放器-图形界面版。  
simplest\_directshow\_info: 最简单的Directshow信息显示例子。  
simplest\_directshow\_filter: 目前还未完成。

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/42394509>

文章标签：[DirectShow](#) [Filter](#) [Pin](#) [播放器](#)

个人分类：[DirectShow](#) [我的开源项目](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com