

## 原 最简单的基于DirectShow的示例：视频播放器图形界面版

2015年01月11日 11:22:30 阅读数：7317

=====

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](#)

=====

本文记录一个最简单的基于DirectShow的图形界面的视频播放器。基于DirectShow的图形界面的播放器的例子还是比较多的，但是大部分都是“层层封装”的例子。“层层封装”的例子相对来说更加稳定，但是却不是很容易理解。因为DirectShow本身的接口函数的数量就比较多，如果再加上封装DirectShow的函数，合起来的函数数量是非常大的，很容易让人搞不清哪些才是真正的DirectShow接口函数。本播放器剥去了DirectShow例子中的“层层封装”，直接调用DirectShow的接口完成视频的播放工作，更加适合DirectShow入门使用。



## 几个功能的实现机制

整个工程的代码比较多，不再详细记录。在这里简单记录一下代码中的几个关键点。

### 视频的播放/暂停/继续/停止

#### 播放

视频“播放”的源代码如下所示。简单来说，完成了以下视频播放的初始化工作：

- (1) 输入的URL转换为Unicode编码（RenderFile()函数支持的输入是Unicode字符串）。
- (2) 调用RenderFile()“智能”创建Filter Graph。
- (3) 调用IMediaControl的Run()方法开始播放视频。
- (4) 开启定时器，用于更新视频播放的进度（后文详细记录）

```

1. void CplayerGUIDlg::OnBnClickedStart()
2. {
3.     CStringA cstr_urla;
4.     CStringW cstr_urlw;
5.     HRESULT hr;
6.
7.     //Render
8.     #ifdef _UNICODE
9.         m_url.GetWindowText(cstr_urlw);
10.    #else
11.        USES_CONVERSION;
12.        m_url.GetWindowText(cstr_urla);
13.        cstr_urlw.Format(L"%s", A2W(cstr_urla));
14.    #endif
15.    if(cstr_urlw.IsEmpty()){
16.        AfxMessageBox(_T("Input URL is NULL!"));
17.        return;
18.    }
19.
20.    hr = pGraph->RenderFile(cstr_urlw, NULL);
21.    if(FAILED(hr)){
22.        AfxMessageBox(_T("Can't open input file!"));
23.        return;
24.    }
25.
26.    //Set Window
27.    HWND screen_hwnd=NULL;
28.    RECT windowRect;
29.    screen_hwnd = this->GetDlgItem(IDC_SCREEN)->GetSafeHwnd();
30.    ::GetClientRect(screen_hwnd, &windowRect);
31.
32.    pWindow->put_Visible(OAFALSE);
33.    pWindow->put_Owner((OAHWND)screen_hwnd);
34.    pWindow->put_Left(0);
35.    pWindow->put_Top(0);
36.    pWindow->put_Width(windowRect.right - windowRect.left);
37.    pWindow->put_Height(windowRect.bottom - windowRect.top);
38.    pWindow->put_WindowStyle(WS_CHILD|WS_CLIPCHILDREN|WS_CLIPSIBLINGS|WS_THICKFRAME);
39.    pWindow->put_MessageDrain((OAHWND) screen_hwnd); //Receive Message
40.    pWindow->put_Visible(OATRUE);
41.
42.    pEvent->SetNotifyWindow((OAHWND)screen_hwnd, WM_GRAPHNOTIFY, 0);
43.
44.    // Run
45.    hr = pControl->Run();
46.
47.    playerstate=STATE_PLAY;
48.    SetBtn(STATE_PLAY);
49.    SetTimer(1,1000,NULL);
50. }

```

## 暂停/继续

视频“暂停/继续”的源代码如下所示。其中调用了IMediaControl的Pause()和Run()设定“暂停”或者是“继续”。

```

1. void CplayerGUIDlg::OnBnClickedPause()
2. {
3.     HRESULT hr;
4.     if(playerstate==STATE_PLAY){
5.         hr=pControl->Pause();
6.         playerstate=STATE_PAUSE;
7.         GetDlgItem(ID_PAUSE)->SetWindowText(_T("Resume"));
8.     }else if(playerstate==STATE_PAUSE){
9.         hr=pControl->Run();
10.        playerstate=STATE_PLAY;
11.        GetDlgItem(ID_PAUSE)->SetWindowText(_T("Pause"));
12.    }
13.
14. }

```

## 停止

视频的“停止”的源代码如下所示。该部分代码完成了以下工作：

- (1)  
把播放的位置重新调整为0
- (2)  
调用IMediaControl的Pause()
- (3)  
关闭定时器
- (4)  
删除Filter Graph中的Filter

```

1. void CplayerGUIDlg::OnBnClickedStop()
2. {
3.     long long position = 0;
4.     HRESULT hr;
5.     hr = pSeeking->SetPositions(&position, AM_SEEKING_AbsolutePositioning | AM_SEEKING_SeekToKeyFrame,
6.         0, AM_SEEKING_NoPositioning);
7.     KillTimer(1);
8.     hr=pControl->Stop();
9.
10.    // Enumerate the filters And remove them
11.    IEnumFilters *pEnum = NULL;
12.    hr = pGraph->EnumFilters(&pEnum);
13.    if (SUCCEEDED(hr))
14.    {
15.        IBaseFilter *pFilter = NULL;
16.        while (S_OK == pEnum->Next(1, &pFilter, NULL))
17.        {
18.            // Remove the filter.
19.            pGraph->RemoveFilter(pFilter);
20.            // Reset the enumerator.
21.            pEnum->Reset();
22.            pFilter->Release();
23.        }
24.        pEnum->Release();
25.    }
26.
27.    SystemClear();
28. }

```

## 视频播放进度在时间轴的显示

随着视频的播放，需要在视频播放进度的时间轴上更新播放进度信息。在程序中使用了一个定时器完成这个功能。在视频开始播放的时候，调用SetTimer()开启定时器。时间间隔设置为1000ms。

```

1. SetTimer(1,1000,NULL);

```

在视频停止播放的时候，调用KillTimer()结束定时器。

```

1. KillTimer(1);

```

在定时器的消息响应函数中，调用了IMediaSeeking的GetCurrentPosition()获取视频当前播放到的时间，调用了IMediaSeeking的GetDuration ()获取视频的时长。根据以上函数得到的数值，计算后把结果设置到相应的控件上。这部分的代码如下所示。

```

1. void CplayerGUIDlg::OnTimer(UINT_PTR nIDEvent)
2. {
3.     if (nIDEvent == 1){
4.         CString curtimestr,durationstr;
5.         long long curtime;
6.         long long duration;
7.         int tns, thh, tmm, tss;
8.         int progress;
9.         //ms
10.        pSeeking->GetCurrentPosition(&curtime);
11.        if(curtime!=0){
12.            //change to second
13.            tns = curtime/10000000;
14.            thh = tns / 3600;
15.            tmm = (tns % 3600) / 60;
16.            tss = (tns % 60);
17.            curtimestr.Format(_T("%02d:%02d:%02d"),thh,tmm,tss);
18.            m_curtime.SetWindowText(curtimestr);
19.        }
20.        pSeeking->GetDuration(&duration);
21.        if(duration!=0){
22.            tns = duration/10000000;
23.            thh = tns / 3600;
24.            tmm = (tns % 3600) / 60;
25.            tss = (tns % 60);
26.            durationstr.Format(_T("%02d:%02d:%02d"),thh,tmm,tss);
27.            m_duration.SetWindowText(durationstr);
28.
29.            progress=curtime*100/duration;
30.            m_progress.SetPos(progress);
31.        }
32.    }
33.    CDialogEx::OnTimer(nIDEvent);
34. }

```

## 视频播放点的调整

当鼠标拖动滑动控制条（Slider Control）控件上的滑块的时候，需要根据拖动的位置设置视频的播放进度。此时调用IMediaSeeking的SetPositions()设定视频的播放进度。消息响应函数中的代码如下所示。

```
[cpp]
1. void CplayerGUIDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
2. {
3.     if (pScrollBar->GetSafeHwnd() == m_progress.GetSafeHwnd()){
4.         float pos_bar=0.0;
5.         long long duration=0.0;
6.         long long pos_time=0.0;
7.         if(nSBCode==SB_THUMBPOSITION){
8.             pos_bar=(float)nPos/100.0;
9.             pSeeking->GetDuration(&duration);
10.            pos_time=pos_bar*duration;
11.
12.            long long position = (long long)(pos_time);
13.            HRESULT hr = pSeeking->SetPositions(&position, AM_SEEKING_AbsolutePositioning | AM_SEEKING_SeekToKeyFrame,
14.                0, AM_SEEKING_NoPositioning);
15.        }
16.    }
17.    CDialogEx::OnHScroll(nSBCode, nPos, pScrollBar);
18. }
```

## “全屏播放”的问题

视频的全屏播放通过IVideoWindow的put\_FullScreenMode()实现，代码如下所示。

```
[cpp]
1. void CplayerGUIDlg::OnBnClickedFullscreen()
2. {
3.     pWindow->put_FullScreenMode(OATRUE);
4. }
```

同时，在“全屏模式”启动后，如果按“ESC”键的话，可以关闭“全屏模式”。这部分的代码在PreTranslateMessage()中实现，如下所示。

```
[cpp]
1. //Exit Full Screen mode when push "ESC"
2. BOOL CplayerGUIDlg::PreTranslateMessage(MSG* pMsg)
3. {
4.     if (pMsg->message == WM_KEYDOWN){
5.         if (pMsg->wParam == VK_RETURN || pMsg->wParam == VK_ESCAPE){
6.             // Restore form fullscreen mode
7.             pWindow->put_FullScreenMode(OAFALSE);
8.
9.             return 1;
10.        }
11.    }
12.    return CDialogEx::PreTranslateMessage(pMsg);
13. }
```

在这里有一点需要注意，IVideoWindow的put\_FullScreenMode()在Win7下是有问题的。只有在设置窗口样式的的时候，在样式中指定WS\_THICKFRAME后才可以正常使用。例如如下代码。

```
[cpp]
1. pWindow->put_WindowStyle(WS_CHILD|WS_CLIPCHILDREN|WS_CLIPSIBLINGS|WS_THICKFRAME);
```

如果没有指定WS\_THICKFRAME样式的话，在退出“全屏”模式之后，视频就显示不出来了，取而代之的是一片黑色。

但是设定WS\_THICKFRAME样式之后，视频窗口的外围会有一层“白边”，会影响到视频显示的美观。因此我们如果希望正常使用全屏的话，可能需要找一种更好的方法，在这里我就没有深入研究了。

## 运行结果

这是使用DirectShow基于MFC开发的一个示例播放器。实现了一个播放器的基本功能：播放，暂停/继续，停止，播放时间轴的显示，以及从任一点开始播放媒体。并且支持将媒体文件拖拽至播放器进行播放。播放前将媒体文件的路径输入到“URL”栏中，然后单击“Start”即可开始播放。在软件下方包含了“start”，“Pause”，“Stop”等按钮用于控制媒体的播放。



播放时候的效果截图如下所示。



单击“Full Screen”可以全屏播放。单击“Info”可以显示正在播放媒体的信息，包括以下两种信息：

- (1)  
该视频的相关信息
- (2)  
播放该视频的 Filter Graph 中的 Filter。



## 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/42372631>

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

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