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：7337

==========

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）

```cpp
void CplayerGUIDlg::OnBnClickedStart()
{
    CStringA cstr_urla;
    CStringW cstr_urlw;
    HRESULT hr;

    //Render
#ifdef _UNICODE
    m_url.GetWindowText(cstr_urlw);
#else
    USES_CONVERSION;
    m_url.GetWindowText(cstr_urla);
    cstr_urlw.Format(L"%s",A2W(cstr_urla));
#endif
    if(cstr_urlw.IsEmpty()){
        AfxMessageBox(_T("Input URL is NULL!"));
        return;
    }

    hr = pGraph->RenderFile(cstr_urlw, NULL);
    if(FAILED(hr)){
        AfxMessageBox(_T("Can't open input file!"));
        return;
    }

    //Set Window
    HWND screen_hwnd=NULL;
    RECT windowRect;
    screen_hwnd = this->GetDlgItem(IDC_SCREEN)->GetSafeHwnd();
    ::GetClientRect(screen_hwnd, &windowRect);

    pWindow->put_Visible(OAFALSE);
    pWindow->put_Owner((OAHWND)screen_hwnd);
    pWindow->put_Left(0);
    pWindow->put_Top(0);
    pWindow->put_Width(windowRect.right - windowRect.left);
    pWindow->put_Height(windowRect.bottom - windowRect.top);
    pWindow->put_WindowStyle(WS_CHILD|WS_CLIPCHILDREN|WS_CLIPSIBLINGS|WS_THICKFRAME);
    pWindow->put_MessageDrain((OAHWND) screen_hwnd);//Receive Message
    pWindow->put_Visible(OATRUE);

    pEvent->SetNotifyWindow((OAHWND)screen_hwnd, WM_GRAPHNOTIFY, 0);

    // Run
    hr = pControl->Run();

    playerstate=STATE_PLAY;
    SetBtn(STATE_PLAY);
    SetTimer(1,1000,NULL);
}
```

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”。

```cpp
void CplayerGUIDlg::OnBnClickedPause()
{
    HRESULT hr;
    if(playerstate==STATE_PLAY){
        hr=pControl->Pause();
        playerstate=STATE_PAUSE;
        GetDlgItem(ID_PAUSE)->SetWindowText(_T("Resume"));
    }else if(playerstate==STATE_PAUSE){
        hr=pControl->Run();
        playerstate=STATE_PLAY;
        GetDlgItem(ID_PAUSE)->SetWindowText(_T("Pause"));
    }

}
```

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

```cpp
void CplayerGUIDlg::OnBnClickedStop()
{
    long long position = 0;
    HRESULT hr;
    hr = pSeeking->SetPositions(&position, AM_SEEKING_AbsolutePositioning | AM_SEEKING_SeekToKeyFrame,
        0, AM_SEEKING_NoPositioning);
    KillTimer(1);
    hr=pControl->Stop();

    // Enumerate the filters And remove them
    IEnumFilters *pEnum = NULL;
    hr = pGraph->EnumFilters(&pEnum);
    if (SUCCEEDED(hr))
    {
        IBaseFilter *pFilter = NULL;
        while (S_OK == pEnum->Next(1, &pFilter, NULL))
        {
            // Remove the filter.
            pGraph->RemoveFilter(pFilter);
            // Reset the enumerator.
            pEnum->Reset();
            pFilter->Release();
        }
        pEnum->Release();
    }

    SystemClear();
}
```

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。

```cpp
SetTimer(1,1000,NULL);
```

在视频停止播放的时候，调用KillTimer()结束定时器。

```cpp
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。

```cpp
void CplayerGUIDlg::OnTimer(UINT_PTR nIDEvent)
{
    if (nIDEvent == 1){
        CString curtimestr,durationstr;
        long long curtime;
        long long duration;
        int tns, thh, tmm, tss;
        int progress;
        //ms
        pSeeking->GetCurrentPosition(&curtime);
        if(curtime!=0){
            //change to second
            tns = curtime/10000000;
            thh  = tns / 3600;
            tmm  = (tns % 3600) / 60;
            tss  = (tns % 60);
            curtimestr.Format(_T("%02d:%02d:%02d"),thh,tmm,tss);
            m_curtime.SetWindowText(curtimestr);
        }
        pSeeking->GetDuration(&duration);
        if(duration!=0){
            tns = duration/10000000;
            thh  = tns / 3600;
            tmm  = (tns % 3600) / 60;
            tss  = (tns % 60);
            durationstr.Format(_T("%02d:%02d:%02d"),thh,tmm,tss);
            m_duration.SetWindowText(durationstr);

            progress=curtime*100/duration;
            m_progress.SetPos(progress);
        }
    }
    CDialogEx::OnTimer(nIDEvent);
}
```

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
void CplayerGUIDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    if (pScrollBar->GetSafeHwnd() == m_progress.GetSafeHwnd()){
        float pos_bar=0.0;
        long long duration=0.0;
        long long pos_time=0.0;
        if(nSBCode==SB_THUMBPOSITION){
            pos_bar=(float)nPos/100.0;
            pSeeking->GetDuration(&duration);
            pos_time=pos_bar*duration;

            long long position = (long long)(pos_time);
            HRESULT hr = pSeeking->SetPositions(&position, AM_SEEKING_AbsolutePositioning | AM_SEEKING_SeekToKeyFrame,
                0, AM_SEEKING_NoPositioning);
        }
    }
    CDialogEx::OnHScroll(nSBCode, nPos, pScrollBar);
}
```

“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
void CplayerGUIDlg::OnBnClickedFullscreen()
{
    pWindow->put_FullScreenMode(OATRUE);
}
```

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
//Exit Full Screen mode when push "ESC"
BOOL CplayerGUIDlg::PreTranslateMessage(MSG* pMsg)
{
    if (pMsg->message == WM_KEYDOWN){
        if (pMsg->wParam == VK_RETURN || pMsg->wParam == VK_ESCAPE){
            // Restore form fullscreen mode
            pWindow->put_FullScreenMode(OAFALSE);

            return 1;
        }
    }
    return CDialogEx::PreTranslateMessage(pMsg);
}
```

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
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。

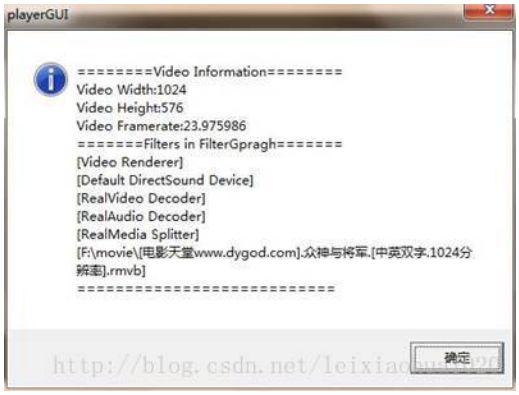
单击“Full Screen”可以全屏播放。单击“Info”可以显示正在播放媒体的信息，包括以下两种信息：

（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

开源中国： http://git.oschina.net/leixiaohua1020/simplest_directshow_example

CDSN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。

文章标签： DirectShow 播放器 MFC
个人分类： DirectShow 我的开源项目