

change avi source files in real-time

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.video/2007-02/msg00>

- *From:* john <john@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 6 Feb 2007 21:05:01 -0800
-

this is just texture3d9 sample.

this sample loads "wFileName" and play it.

I guess

```
" hr = g_pGB->AddSourceFilter (wFileName, L"SOURCE", &pFSrc); "
```

this is loading part.

I have a question now.

I want to stop current AVI and want to load new AVI file.
and play it.

where should I modify in this source code ?

and another question is
if I want to play 7 avi files sequently ,
if I make 7 g_pGB instances and play it, it have problem ?

```
HRESULT InitDShowTextureRenderer()  
{  
    HRESULT hr = S_OK;  
    CComPtr<IBaseFilter> pFSrc; // Source Filter  
    CComPtr<IPin> pFSrcPinOut; // Source Filter Output Pin  
    CTextureRenderer *pCTR=0; // DirectShow Texture renderer  
  
    // Create the filter graph  
    if (FAILED(g_pGB.CoCreateInstance(CLSID_FilterGraph, NULL,  
    CLSCTX_INPROC)))  
        return E_FAIL;  
  
    #ifndef REGISTER_FILTERGRAPH  
    // Register the graph in the Running Object Table (for debug purposes)  
    AddToROT(g_pGB);  
    #endif
```

change avi source files in real-time

```
// Create the Texture Renderer object
pCTR = new CTextureRenderer(NULL, &hr);
if (FAILED(hr) || !pCTR)
{
    Msg(TEXT("Could not create texture renderer object! hr=0x%x"), hr);
    return E_FAIL;
}

// Get a pointer to the IBaseFilter on the TextureRenderer, add it to
graph
g_pRenderer = pCTR;
if (FAILED(hr = g_pGB->AddFilter(g_pRenderer, L"TEXTURERENDERER")))
{
    Msg(TEXT("Could not add renderer filter to graph! hr=0x%x"), hr);
    return hr;
}

// Determine the file to load based on DirectX Media path (from SDK)
// Use a helper function included in DXUtils.cpp
TCHAR strFileName[MAX_PATH] = {NULL,};
WCHAR wFileName[MAX_PATH] = {NULL,};

//Istrcpy( strFileName, DXUtil_GetDXSDKMediaPath(), MAX_PATH-1 );
//Istrcat( strFileName, SOURCE_FILE );
Istrcpy( strFileName, SOURCE_FILE );
strFileName[MAX_PATH-1] = 0; // NULL-terminate
wFileName[MAX_PATH-1] = 0; // NULL-terminate

USES_CONVERSION;
wcsncpy(wFileName, T2W(strFileName), NUMELMS(wFileName));

// Add the source filter to the graph.
hr = g_pGB->AddSourceFilter (wFileName, L"SOURCE", &pFSrc);

// If the media file was not found, inform the user.
if (hr == VFW_E_NOT_FOUND)
{
    Msg(TEXT("Could not add source filter to graph!
(hr==VFW_E_NOT_FOUND)\r\n\r\n")
    TEXT("This sample reads a media file from the DirectX SDK's
media path.\r\n")
    TEXT("Please install the DirectX 9 SDK on this machine."));
    return hr;
}
else if(FAILED(hr))
{
    Msg(TEXT("Could not add source filter to graph! hr=0x%x"), hr);
    return hr;
}
```

change avi source files in real-time

```
if (FAILED(hr = pFSrc->FindPin(L"Output", &pFSrcPinOut)))
{
Msg(TEXT("Could not find output pin! hr=0x%x"), hr);
return hr;
}

#ifdef NO_AUDIO_RENDERER

// If no audio component is desired, directly connect the two video pins
// instead of allowing the Filter Graph Manager to render all pins.

CComPtr<IPin> pFTRPinIn; // Texture Renderer Input Pin

// Find the source's output pin and the renderer's input pin
if (FAILED(hr = pFTR->FindPin(L"In", &pFTRPinIn)))
{
Msg(TEXT("Could not find input pin! hr=0x%x"), hr);
return hr;
}

// Connect these two filters
if (FAILED(hr = g_pGB->Connect(pFSrcPinOut, pFTRPinIn)))
{
Msg(TEXT("Could not connect pins! hr=0x%x"), hr);
return hr;
}

#else

// Render the source filter's output pin. The Filter Graph Manager
// will connect the video stream to the loaded CTextureRenderer
// and will load and connect an audio renderer (if needed).

if (FAILED(hr = g_pGB->Render(pFSrcPinOut)))
{
Msg(TEXT("Could not render source output pin! hr=0x%x"), hr);
return hr;
}

#endif

// Get the graph's media control, event & position interfaces
g_pGB.QueryInterface(&g_pMC);
g_pGB.QueryInterface(&g_pMP);
g_pGB.QueryInterface(&g_pME);

// Start the graph running;
if (FAILED(hr = g_pMC->Run()))
{
Msg(TEXT("Could not run the DirectShow graph! hr=0x%x"), hr);
return hr;
}
```

change avi source files in real-time

change avi source files in real-time

}

return S_OK;

}

.