

Re: Using VMR9 outside of DShow?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.video/2004-12/0557.1>

From: Thore Karlsen [MVP DX] (*sid_at_6581.com*)

Date: 12/22/04

Date: Wed, 22 Dec 2004 11:09:38 -0600

On Wed, 22 Dec 2004 08:47:18 -0700, "Jeremy Noring"
<somewhere@thenet.com> wrote:

>> *No, you don't need to do that. You can make it as a regular C++ class
>> and instantiate it using new. Use a smart pointer like this:*
>>
>> *CComPtr<InprocClient> client(new InprocClient(0, &hr));*
>>
>> *That's all there is to it!*

>*Sweet! A few questions:*

>*1. Is "InprocClient" the name of the class I'm instantiating?*

Yes, and actually I forgot to add the I for the interface name for
CComPtr. But you probably would have figured that out when the compiler
complained.

>*2. After instantiating a class like this, I can use that class just like I
>would a regular dshow filter (assuming, of course, that the class was
>originally a dshow filter)?*

Yep, add it to a graph, and use it like any other filter. No difference.

>> *You can actually register things, I used to do that before I moved my
>> filters inside the executables. You just need to set up an .inf file to
>> register it. I don't know the exact details, since I didn't do the
>> ActiveX CAB files. I did write the ActiveX controls, and they works just
>> fine in web pages with filters inside.*

>*So, in your opinion, which would be easier? I need to use custom filters,
>but I can't register them--is it easier to go with an .inf file, or to use
>the smart pointers?*

I'd put the filters inside your control, and skip the separate
files/registration mess. The fewer dependencies you have, the less
headaches. I don't have any filters outside my executables anymore, it
lead to too many problems and support issues.

microsoft.public.win32.programmer.directx.video: Re: Using VMR9 outside of DShow?

In my case the ActiveX control used the same filters our applications used, so if different filters were installed by the ActiveX control and the application... Not so good! COM hell is even worse than regular DLL hell, because unless you jump through hoops you're only supposed to have one version of an object registered on the system.

Also, COM interfaces aren't supposed to change, but that is ridiculous when you only need to use these filters in your own programs and don't have to worry about breaking other programs. I have custom interfaces on my filters that can change quite a bit between versions, and the only foolproof way to make sure that doesn't lead to problems is to keep the filters inside the executables.

--

New to newsgroups? Read: <http://dev.6581.com/newsgroups.html>