

Re: Hardware Overlay with Push Source

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.video/2005-04/msg00>

- *From:* "Alessandro Angeli [MVP::DigitalMedia]" <nobody@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 13 Apr 2005 05:21:39 +0200
-

Augie at NHN wrote:

- > Is it just because XVID decode filter uses the Video Transform instead of the Source Stream?

There is no difference between filters: an upstream filter is an upstream filter whether it has an input pin (transform) or not (source).

This is quoted from the VR docs (my idea is after the quote):

<<<

The Video Renderer accepts YUV formats if the video graphics card supports YUV overlay surfaces. When it first connects to the upstream filter, however, the Video Renderer requires an RGB format that matches the color depth of the current monitor settings. For example, if the current display setting is 24-bit color, the upstream filter must be able to provide 24-bit RGB video. When the filter graph switches to a running state, the Video Renderer negotiates a dynamic format change to the appropriate YUV color space.

By connecting with an RGB type, the Video Renderer ensures that it can use GDI in case DirectDraw is not available. It will switch to GDI if another application is using the video memory, if the video rectangle straddles two monitors on a multi-monitor system, or if the video rectangle is completely obscured by another window.

Note The Video Mixing Renderer does not perform this type of dynamic format change, and does not require an RGB media type, because it never uses GDI for rendering.

To negotiate a format change, the Video Renderer calls IPin::QueryAccept with the new media type. If the upstream filter returns S_OK, the Video Renderer attaches the new media to the next sample. The upstream filter should call

Re: Hardware Overlay with Push Source

IMediaSample::GetMediaType on each sample. If GetMediaType returns a non-NULL value, it indicates a format change, and the upstream filter should respond by switching output types. (Do not switch types in the QueryAccept method.) The upstream filter should accept at least the major RGB types, and ideally should support the common YUV types. During streaming, the Video Renderer might switch back and forth between YUV and RGB types any number of times. The Video Renderer does not accept dynamic format changes initiated by the upstream filter.

When the Video Renderer draws to a DirectDraw overlay surface, it allocates a single buffer for its input pin. If the upstream filter attempts to force a connection using multiple buffers, the Video Renderer will be unable to use the overlay surface.

>>>

So, there may be 2 issues: if the source filter does not use a single buffer or does not support the format change, the VR will not be able to use the overlay.

--

// Alessandro Angeli
// MVP :: Digital Media
// a dot angeli at psynet dot net

- *Follow-Ups:*

- ◆ **Re: Hardware Overlay with Push Source**

- ◇ From: Augie at NHN

- *References:*

- ◆ **Hardware Overlay with Push Source**

- ◇ From: Augie at NHN

- ◆ **Re: Hardware Overlay with Push Source**

- ◇ From: Alessandro Angeli [MVP::DigitalMedia]

- ◆ **Re: Hardware Overlay with Push Source**

- ◇ From: Augie at NHN

- ◆ **Re: Hardware Overlay with Push Source**

- ◇ From: The March Hare [MVP]

- ◆ **Re: Hardware Overlay with Push Source**

- ◇ From: Augie at NHN

Re: Hardware Overlay with Push Source

- Prev by Date: [*How to get IVideoWindow's HWND?*](#)
- Next by Date: [*Re: How to set Fullscreen in VMR9 filter?*](#)
- Previous by thread: [*Re: Hardware Overlay with Push Source*](#)
- Next by thread: [*Re: Hardware Overlay with Push Source*](#)
- Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)