
Subject: Re: Can a CALL_EXTERNAL .dll create a window?
Posted by [Matt Feinstein](#) on Sat, 30 Aug 2003 11:39:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <vkvgj3q36nf1@corp.supernews.com>, Karl Schultz
<kschultz_no_spam@rsinc.com> wrote:

... after some interesting and helpful comments, Karl adds...

> You mention "off-screen hardware assisted rendering". I'm not sure that you
> are going to have much luck here. Most hardware accelerators are wired
> pretty tightly to the video system's frame buffer. I've not heard of many
> hardware accelerators that can render into off-screen memory. Do you want
> to render into a region that is in the video memory, but not in a displayed
> window? I guess I'm not sure how you are going to convert your on-screen
> hardware rendering context into an off-screen one. I'm not sure that
> Microsoft OpenGL will use hardware rendering when drawing to a
> device-dependent bitmap. I know that it reverts to software when writing to
> a DIB.

Actually, you -can- do this with the pbuffer extension, which is an ARB extension to the wgl functions and is supported by all the graphics board vendors. There is also, I think, an analogous glX extension. It's true that if you use the 'DRAW_TO_BITMAP' pixel format property, then, as you say, you get just the OpenGL 1.1 software renderer. But the pbuffer extension lets you create an off-screen hardware accelerated context with whatever pixel format you want (within limits). One of the pbuffer extension functions lets you query the 'acceleratedness' of the pbuffer, and you can confirm that it is, indeed, hardware-accelerated. And it really does work-- the only catch, as I mentioned in my first post, is that you have to start with an on-screen context.

Matt
