Subject: Re: Backing Store Question
Posted by Karl Schultz on Fri, 16 Sep 2005 19:25:02 GMT
View Forum Message <> Reply to Message

On Fri, 16 Sep 2005 07:35:53 -0600, David Fanning wrote:

> Folks,
>
> In poking around in the IDL documentation this morning, I came
> upon this in the WIDGET_DRAW documentation:
>
> On some systems, when backing store is provided by the
> window system (RETAIN=1), reading data from a window
> using TVRD( ) may cause unexpected results. For example,
> data may be improperly read from the window even when
> the image displayed on screen is correct. Having IDL
> provide the backing store (RETAIN=2) ensures that the
> window contents will be read properly.
>
> Anybody have any idea which systems this is taking about?
>

It is referring to X servers. On Windows, setting RETAIN to 1 effectively sets RETAIN to 0.

The intent behind RETAIN=1 in the X environment is to use the X server's backing store to repair the window when it needs repairing. The idea is that when a window is damaged, the X server tries to repair it itself from bits it may have stored in backing store. If the server can't do it, it will send an Expose event to the application, which means running the IDL application's expose event hander, if there is one. If you didn't code an expose handler, hoping that RETAIN=1 would always repair the window for you, then you'll be out of luck if the X backing store mechanism fails for some reason.

The server may be able to repair with backing store at one point in time, and then may fail to do so later if resources (memory) get too tight. So, the entire backing store functionality is a little bit unsteady in this respect. I think that backing store may have been more popular and important in older systems when it took longer to render things. Nowadays, I'm a little hard-pressed to find X servers that can be configured to use backing store. I know that some of the latest Xorg servers do not support it.

I suppose that the warning about the TVRD is pretty much the same issue as being able to repair the window. The server could respond to a read request with pixels from the backing store, but if it had to give up on providing the backing store, it wouldn't be able to give back those

pixels. I'm a little confused about why reading them off an undamaged window wouldn't work, but I think that's the general idea.

Karl
