
Subject: Re: retain and graphics_level=2
Posted by [davidf](#) on Wed, 29 Oct 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Stein Vidar Hagfors Haugan (s.v.h.haugan@astro.uio.no) writes
in follow-up to this discussion:

>> Like everything else, it depends. On my non-OpenGL-
>> accelerated machines I find it noticeably slower, but
>> not unbearably slow for most object graphics programs.
>> It obviously depends on how complicated the graphics
>> scene is to render.
>
> This is where I don't follow - if the graphics object is
> *not* changed it shouldn't have to be rendered over again.

I think you are confusing the word "create" or "change" with "render". Graphic objects, it is true, do not have to be re-created each time they are used. And they can be changed at will. But in order to see changes in effect, the scene must be re-rendered or displayed in the window. This is usually done by invoking the Draw method on the Window object. Graphics windows must be re-rendered when part of the window needs to be repaired. (The technical term for this is "backing store".) A window must be repaired, for example, if it is uncovered by moving a window that was in front of it.

In direct graphics either the window manager (Retain=1) or IDL (Retain=2) keeps a pixmap of the window to effect this kind of window repair. This is absolutely necessary in direct graphics because once a direct graphics command is issued there is absolutely no record of it. In other words, the window doesn't "know" anything about what is in it. But in object graphics, the window can in some sense be said to "know" about its contents. Thus, if a window object needs to be repaired, it makes sense for the object to repair itself. To have IDL do it (which, as I say, doesn't work) or to have the window system do it (which it can't do on my PC) would be overkill.

> Another issue is that some programs may decide to render
> the graphics scene, then do changes to the graphics object
> as the result of a *series* of events, waiting for a
> push on a "Render" button to show the effects. If the
> window is hidden/exposed in the intermediate state, then

- > the automatic call to the Draw method would erroneously
- > display the changes straight away.... Ok, I agree the
- > example is somewhat far-fetched, but still.

Well, just turn off Expose events on the Draw widget while you fidget about. :-)

- >> I have learned, by the way, that speeding up object
- >> graphics is a high priority for the folks at RSI in
- >> their next release of IDL.
- >
- > I would much rather that they (first, at least) focus on making
- > sure that widgets that worked under 4.0.1 will work under 5.X
- > (e.g., the "multiple UPDATE on/off" problem that masks out
- > the contents of scrollable sub-bases).

Yes, well, perhaps getting widgets to work properly is their *highest* priority. I hope so, too.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
