

---

Subject: Re: IDLgrWindow vs. IDLgrBuffer, font-rendering difference

Posted by [Karl\[1\]](#) on Mon, 30 Nov 2009 16:47:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Nov 30, 9:41 am, David Fanning <n...@dfanning.com> wrote:

> M. Katz writes:

>> On Mac OS X I'm using 1.45 as the scaling factor (i.e. I make the

>> IDLgrBuffer fonts larger by 1.45x to match IDLgrWindow). I have not

>> tested Windows, but David says they match.

>

> What has ITTVIS had to say about this? I was on their

> web page this morning looking for news, but didn't find

> anything in the Tech Tips. I'm never sure, though, if this

> is because there is nothing there or because their hapless

> search engine is still inept. In any case "IDLgrBuffer"

> or "buffer" or "object graphics" didn't turn anything

> up.

>

> Cheers,

>

> David

>

> --

> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

I think it has to do with the default RESOLUTION of the IDLgrBuffer device.

Recall that an IDLgrBuffer is not a physical device. Since there is no piece of hardware for it to query, it simply sets its RESOLUTION property to 72 DPI as a somewhat arbitrary, but reasonable, default value. Since this property is writable, you can change it.

An IDLgrWindow can query the hardware and figure out how big a pixel is (resolution). It then sets its RESOLUTION property accordingly to communicate the value to the application; it cannot be changed.

If text is to be drawn at a specific physical size (e.g., 12 points or 12/72 inches), then the object needs to know the size of a pixel in order to draw something of a specific size. On a 72 DPI device, a 12-point font would occupy 12 pixels in height. Of course, this is the size of the entire "box" for the glyph; this does not mean that all glyphs are 12 pixels tall.

You can specify coordinates in Object Graphics using inches and

centimeters, right? If OG is going to draw a line 1 inch long, it had better know the size of pixels on the device so it knows how many to draw to make an inch.

The first post to this thread mentioned a 4/3 scale factor. Many displays are 96 DPI, which explains that ratio. Perhaps the 1.45 comes from Apple display hardware.

It seems to me that the smart way to fix this is to query the RESOLUTION from the IDLgrWindow that you want to match up with the IDLgrBuffer and set the IDLgrBuffer's RESOLUTION property to match. This should be better than scaling the font size with some fixed value. It will also work with displays that have different resolutions than the ones used to compute this fixed scale factor. Other things in Object Graphics like line widths use the RESOLUTION property, as well as things specified in inches or centimeters.

It's been awhile and I'm not sure that this is right. But the docs seem to support the concept. See the reference pages for IDLgrBuffer.

Karl

---