
Subject: Re: Coyote's BIG Idea

Posted by [David Fanning](#) on Mon, 24 Jan 2011 18:30:16 GMT

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

Gray writes:

> Hey David, this may not be a question you can answer, but I'd like to
> know it: what can NG do that CG (coyote graphics) can't, and vice
> versa?

Well, that's a good question. I guess we are still trying to figure that out. :-)

I'm not extremely familiar with the new function graphics routines, because I can't keep IDL 8 running on my computer (Windows 7, 64-bit OS). But I would say one thing Coyote Graphics can't do yet (but probably because I haven't gotten around to trying) is save the graphics commands in a display window and restore them. In fact, the graphics "commands" are objects in the FSC_Window display window, so I suspect I can do this quite easily when I get around to it.

One thing I know for sure the new function graphics can't do that Coyote Graphics can is create a new type of graphics command in a widget program. My challenge to demonstrate how one would build a "window-level" program with function graphics goes unanswered to this day. I strongly suspect this is because it can't be done. At least not by mortals. Maybe someone at ITTVIS could figure it out. But, I don't think, frankly, it is possible to use function graphics in a widget program that requires any kind of user interaction with the graphics window. So no XStretch, no WindowImage, no FSC_Zplot, etc, etc.

The reason I think function graphics is limited in this area is that the graphics functions are very high level routines. Coyote Graphics routines are much lower-level. You can build things with Coyote Graphics, while you have to settle for what's already been built with function graphics.

Function graphics allows you to set graphical "properties" with property sheets. I have something similar in mind, but haven't implemented it yet. I think Coyote Graphics at the moment is closer to how people have traditionally worked in IDL, rather than introducing a whole new way of working and interacting with graphics programs. That is to say, they work as well from the command line as they do in a program. Maybe this is a wash.

I personally think function graphics commands, under the hood,

are so complicated that they will never be bug free. I think there is a very good chance that every time you try to fix something, you will break something else. Coyote Graphics are probably more complicated than I thought they would be, but mostly because I have tried to preserve as much backward compatibility with traditional graphics commands as possible. If I had decided to start from scratch, they would be very, very simple.

I don't know. Maybe other people have some ideas. I suppose it is no secret that I just consider it a challenge to build IDL 8 like programs for IDL 6 like people. Maybe no one will use the programs I write. In truth, that's not something I lose a lot of sleep thinking about. It's fun, it's challenging, and I am blown away by what is possible with old, familiar tools. Good enough for me. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
