
Subject: Resizable Graphics Windows for Traditional Commands

Posted by [David Fanning](#) on Wed, 19 Jan 2011 05:56:25 GMT

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

Folks,

Some of you know I have been engaged the past several months writing a book about traditional graphics commands. It will be ready shortly. As I have written the book, I have been putting ideas from the book into practice in a series of programs I've started to call Coyote Graphics:

http://www.idlcoyote.com/graphics_tips/coyote_graphics.html

There was one program left to finish, and that was a resizable graphics window to display these routines. I spent the past two days completely re-writing the old FSC_Window program from scratch. The old program allowed a single command to be added to a resizable graphics window.

The new program is significantly more powerful. In fact, an unlimited number of graphics "commands" can be added to the window, commands can be deleted and replaced by other commands, listed, and so on. Plus, you can have multiple windows on the display, and you can interact with the commands in any one of the windows.

It is even possible to display multiple plots in the graphics window with a mechanism that will remind you strongly of !P.Multi.

Contents of the graphics window can be sent directly to a PostScript file (you will have to write your own routines to be PostScript compatible, but all the Coyote Graphics routines are already set up for this), or you can save the graphics window in any of five different raster file formats. If you have ImageMagick installed on your computer, you will have the additional option of creating these raster files from PostScript files, which dramatically improves the quality of the raster output, especially fonts.

You can read more about FSC_Window in this article:

http://www.idlcoyote.com/graphics_tips/fsc_window.html

This is really a very easy way to create graphical output

in a resizable graphics window in IDL. You will be able to run this program in any version of IDL, as far as I know.

I think we have about 2-3 weeks to play with this program before the final code freeze for the book. Please let me know if you have any comments.

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.")

Subject: Re: Resizable Graphics Windows for Traditional Commands

Posted by [David Fanning](#) on Thu, 20 Jan 2011 15:56:07 GMT

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

David Fanning writes:

> Humm. The first "plot" command definitely works on my Windows
> machine. I'm not really doing anything with the commands
> except "executing" them with Call_Procedure. I can't think
> of a single reason why they wouldn't work everywhere.
>
> I guess I'll fire up the ol' LINUX machine and have
> a look. Strange. :-(

OK, here is the problem. For some reason on UNIX machines the contents of the graphics buffer are not getting flushed when you execute a graphics command with Call_Procedure. I tried putting a FLUSH command at the end of the execution and that didn't help. I tried an EMPTY, and that fixed the problem. (I realized this was the problem, because if I stepped through the code, everything worked perfectly!)

I believe I have seen this problem before. At least I remember stepping through code and having things work, but then NOT working when I just ran the program. I probably should make a note of this. This is probably not the last time we will run into it.

I'll have a new version up in about a half hour. If you want to fix it yourself, put the EMPTY command at the end of the FSC_WINDOW_COMMAND::DRAW method.

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.")

Subject: Re: Resizeable Graphics window
Posted by [Jeremy Bailin](#) on Thu, 17 Feb 2011 19:44:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

You could try doing /loadcmd instead of /addcmd, and then after the for loop execute them with
cgwindow, /executecmd

-Jeremy.

Subject: Re: Resizeable Graphics window
Posted by [David Fanning](#) on Thu, 17 Feb 2011 19:47:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

wlandsman writes:

> I am unclear how to add graphics commands to a Coyote graphics window. If I create a simple plot in a resizable window:
>
> cgplot,indgen(100),/window,psym=2
>
> and now I want to add error bars.
>
> for i=0,89 do cgplots,[i,i],[i-1,i+1],/addcmd
>
> I get the plot I want but only after 90 flashes of the window. (If I don't include the /addcmd keyword then the error bars go to a different window.)

I would turn "execution" of the commands off until you were finished loading the commands:

```
cgplot,indgen(100),/window,psym=2
cgControl, EXECUTE=0
for i=0,89 do cgplots,[i,i],[i-1,i+1],/addcmd
cgControl, EXECUTE=1
```

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.")

Subject: Re: Resizeable Graphics window
Posted by [wlandsman](#) on Thu, 17 Feb 2011 19:57:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, February 17, 2011 2:44:06 PM UTC-5, Jeremy Bailin wrote:

```
> You could try doing /loadcmd instead of /addcmd, and then after the for loop execute them with
>
> cgwindow, /executecmd
>
```

Thanks! That works but /LOADCMD (unlike ADDCMD) is not directly available in CGPLOT. So if I want to write a program with optional output to a resizeable window it would look like this:

```
pro myplot>window= window
;/WINDOW - if set, then display in a resizeable graphics window
```

```
cgplot,indgen(100),window = window,psym=2
if keyword_set(window) then begin
  for i=0,89 do cgwindow,'cgplots',[i,i],[i-2,i+2],/loadcmd
  cgwindow,/execute
endif else for i=0,89 cgplots,[i,i],[i-2,i+2]
```

```
return
end
```

Subject: Re: Resizeable Graphics window
Posted by [David Fanning](#) on Thu, 17 Feb 2011 20:03:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

wlandsman writes:

> Thanks! That works but /LOADCMD (unlike ADDCMD) is not directly available in CGPLOT.
> So if I want to write a program with optional output to a resizable window it would look like this:

```
>  
> pro myplot,window= window  
> ;/WINDOW - if set, then display in a resizable graphics window  
>  
> cgplot,indgen(100),window = window,psym=2  
> if keyword_set(window) then begin  
>   for i=0,89 do cgwindow,'cgplots',[i,i],[i-2,i+2],/loadcmd  
>   cgwindow,/execute  
> endif else for i=0,89 cgplots,[i,i],[i-2,i+2]  
>  
>  
> return  
> end
```

Yes, that's another way to do it. I balked at adding yet another keyword (LOADCMD) to each of the Coyote Graphics routines, because I could see there could easily be no end to it, and it was already getting cumbersome. My goal in writing these routines is simplicity, above all.

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.")

Subject: Re: Resizable Graphics window
Posted by [Jeremy Bailin](#) on Thu, 17 Feb 2011 20:24:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
> I would turn "execution" of the commands off until  
> you were finished loading the commands:  
>  
> cgplot,indgen(100),/window,psym=2  
> cgControl, EXECUTE=0  
> for i=0,89 do cgplots,[i,i],[i-1,i+1],/addcmd  
> cgControl, EXECUTE=1
```

Ah yes, that's clearly a better option. :-)= Lots of nice little tricks within CG still to learn...

Speaking of which, I've got another odd issue with the aspect ratio of output. If I output a postscript file, I get the wrong aspect ratio, but if I out a .png file via ImageMagick, I get the right aspect ratio. Even weirder is that if I set ps_delete=0, the intermediate postscript file has the *right* aspect ratio. I've looked at the code in both sections, and I can't see any difference that could account for it... and if I try to manually run the commands that create the ImageMagicked .png file, I get the wrong aspect ratio!

So I'm befuddled. :-)=

{ x86_64 darwin unix Mac OS X 7.0.4 Sep 3 2008 64 64}

-Jeremy.
