Phillip Bitzer writes:

> I'm a big fan of Coyote's routines, especially since I'm still in IDL 7.1 land. However, I'm running into this problem when using them:
>
> cgPlot, dist(10), XTICK_GET=xTickVals
> help, xTickVals
> ----> XTICKVALS   DOUBLE   = Array[6]
>
> No problem there. But, when I want to put this into a cgWindow:
>
> delVar, xTickVals
> cgWINDOW
> cgPlot, dist(10), XTICK_GET=xTickVals, /ADD
> help, xTickVals
> --->XTICKVALS   UNDEFINED = <Undefined>
>
> Digging a little into the code, I find that "extra" in cgPlot has a XTICK_GET, but once I step into cgWINDOW, the "extra" structure no longer has a tag for XTICKVALS. Is this a problem with how I'm using CG? I'll admit I don't totally understand if I should use /WINDOW or /ADD in the call to cgPlot, but I don't think it matters here, since it's the first command.
>
> Has anyone else run into this? Or have any ideas around it? I could just put the output of the initial cgPlot call into a pixmap, but I was hoping I was just missing something.

You are not missing anything. This is one of the true limitations of the Coyote Graphics system. Well, not so much a limitation of Coyote Graphics, but a limitation of the Call_Procedure routine resizeable windows (e.g, cgWindow) use to "replay" the commands inside them when the window is resized.

Call_Procedure just simply doesn't allow you to obtain output keywords in this way. Thus, I have no way of getting this information back to the user or to cgWindow, where I can take advantage of it. Sorry.

If the information doesn't have to be dynamic, you can simple call the routine once (in a pixmap window, if you like) to obtain the information, then save it so you can use it in your program. That's the only way I know to get around the problem.

Cheers,

David

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