Subject: Surprising Odds and Ends
Posted by davidf on Mon, 14 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Folks,

I've been working on the 2nd Edition of my book for the past couple of weeks. As always when I try to write about something in IDL that I think I know something about, I'm learning all kinds of new things. I thought I might share a couple of the stranger things with you. There is no purpose, really, except to get you ready for the IDL Expert Programmers Association exam that will be given early next month to Ben and a couple of other lucky programmers.

1. The units for the TICKLEN keyword are in normalized coordinates relative to the window size. The units for the XTICKLEN and YTICKLEN keywords are in normalized units with respect to the TICKLEN keyword. This makes sense, I suppose, but it is completely contrary to the IDL documentation for XTICKLEN and provided many confusing moments for me. (I actually think I saw this documented somewhere, but I can't find it now when I am looking in all the obvious places.)

2. The HISTOGRAM function documentation states that it will use the minimum and maximum value of the data to calculate image histograms. But this appears to be untrue. At least with byte images, the minimum and maximum values of the histogram are always 0 and 255, no matter what values are in the image.

3. The HEAP_GC command (which, heaven forbid, you *really* shouldn't be using anyway) is dependent on program level. For example, in Cleanup routines I want to destroy pointers in my info structure. But if the program crashes in an event handler, the info structure is undefined in the Cleanup routine. In such a case I might want to clean up the pointers by issuing a HEAP_GC command. But I could never get this to work. Today I found out why. The heap apparently exists *only* at the main IDL level. If you try to call Heap_GC from some other level (e.g., inside a Cleanup routine) the command appears to work, but nothing really happens. This command can *only* be used from the IDL command line.

4. Another way of doing this:

ptrToUndefinedVar = Ptr_New(/Allocate_Heap)

is to do this:

ptrToUndefinedVar = Ptr_New(xx)

where XX is an undefined variable. Neat! It makes storing the extra keyword collect via keyword inheritance MUCH easier.

ptrToExtraKeywords = Ptr_New(extra)

And I always have a valid pointer that can be de-referenced. (I'm sure the pointer gurus already knew this, but it takes me a bit longer sometimes.)

5. PRINTER offsets are calculated from the edge of the printable portion of the page, rather than from the page edge, as they are for PostScript files. Each offset point is different for each printer. Thus, you have to program in printer-specific fudge factors if you want centered output. Ouch!

Enough for now. I'm leaning about 10 new things a day, so I'll probably have more before the exam.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155