Subject: Re: Named Indices
Posted by davidf on Wed, 27 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astro.cornell.edu) writes:

> A slightly more portable solution, in the sense that it requires no modification
> of startup code but simply inclusion of a program somewhere on the path
> (presumably in the same directory tree as the program which would use it), is a
> special purpose function. E.g.
>
> plot, findgen(11), LINESTYLE=linestyle(/DASH)
>
> You could even make an uber-function that encapsulates all such "named
> entity"<->"integer value" constant mappings:
>
> const(/LINESTYLE,/DASH) ; produces 2
>
> const(/AXIS_STYLE,/FORCE,/SUPPRESS) ; produces 4 or 2 = 6
>
> const(/SIZE,/INTEGER) ; produces 2
>
> const(/WIDGET_DRAW_TYPE,/BUTTON_RELEASE) ; produces 1
>
> const(/AXIS_NUMBER,/X) ; produces 0
>
> const(/COLOR_TABLE,/PRISM) ; produces 6
>
> etc.
>
> The problem is you have to maintain it with new versions of IDL, and ensure
> backwards compatibility too. Other issues crop up, such as private color table
> lists, etc. It would definitely help readability though. Sounds like a great
> beginner programming project that would readily educate one on the subtle
> distinctions among keyword_set(), n_elements() ne 0, and arg_present().

I thought about a function too, although not such
a generic one. But I got this far:

```
FUNCTION LS, Solid=Solid, Dot=Dot, Dash=Dash, DashDot=DashDot...
```

When I realized I was in trouble with keyword names.
In the end, I decided cryptic numbers were better
than a long-winded explanation of how keyword names
work. :-(

But what I'm really surprised you didn't offer was
a way to write a polymorphic object to do the job. :-)

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

---