Subject: Re: array dimensions
Posted by John-David T. Smith on Mon, 09 Apr 2001 18:09:37 GMT

Paul van Delst wrote:
>
> JD Smith wrote:
>>
>> Paul van Delst wrote:
>>>
>>> Martin Schultz wrote:
>>>>
>>>> Not a favorite keyword, but a favorite tool: Get a decent system (if
>>>> you haven't one), install emacs with idlwave, run the routine-info
>>>> collection, and then you have all the keywords you never knew but
>>>> always dreamt about literally at your fingertips.
>>>
>>> Similar to typing "?" at the IDL prompt and getting similar info - with examples no less!
>>> Gasp!

OK, just to make the distinction crystal clear: With the RSI online
help, you can search through the index, by typing "size" for example.
This (sometimes) puts you on the routine in question. Hmm, in this
case, it pulls up "SIZE Executive Command". Luckily "SIZE Function" is
the next entry. Try "window" though. Hmm... window object, color
model..., WINDOW system variable, etc. In any case, once you've
(finally) targetted in on your routine of choice, you open up the help
topic (e.g. for WINDOW Procedure) by double-clicking. Sometimes it
opens right away. In this case (and many others), it wants more
information. Do I really want "WINDOW", or perhaps I'm interested in
"Window Systems". OK, "WINDOW" it is. Finally the routine
documentation opens and I can scroll through manually looking for
keywords.

IDLWAVE does all these operations for you instantaneously. If you're
sitting on:

 window,PIXMAP=

and invoke help, IDLWAVE *knows* you're not looking for the !D.WINDOW
system variable, or random notes about window objects, or windowing
functions, or window washing, etc. It knows you'd like some info on the
"PIXMAP" keyword to the "WINDOW" procedure, and dials directly and
instantaneously to:

    PIXMAP
    Set the PIXMAP keyword to specify that
    the window being created is actually an

invisible portion of the display memory called a pixmap.

You get the info you need on the routine/procedure/keyword/system_variable at hand, and get on with your work. I invoke help this way so much now I don't even notice it really. It has become an integral part of my coding technique. As someone who spent many an hour doing it the RSI way with their help browser (type into index name, (possibly) scroll down nearby because it found the wrong thing, double-click, (possibly) dismiss another pop-up, scroll down looking for keywords, etc.), I can guarantee you the IDLWAVE way is *much* faster. It may not be for everyone, but it sure saves me lots of time. One consideration before you throw out the online manuals: IDLWAVE's version is text only, so those cool formatted mathematics and graphics don't make it through very well. This is not a problem 99.5% of the time. I keep RSI's browser handy for the other 0.5%.

I can get at help in many ways: through a keyword completion list, through the routine info, by C-S right clicking on anything (keyword, routine, system variable, object, obj_new invocation, etc.) in the buffer *or* shell, etc.

And here's the best. Lest we not forget some of the best IDL routines are not supplied by RSI... if there is no system help topic for a routine you invoke help on, but its source code is available, that will be loaded up instead. If the source code has a standard documentation header, it will be visited for keywords (with a keyword section), and syntax too. E.g. the nasa procedure, FITS_WRITE:

IDL> fits_write, NANVALUE=

Invoke help... presto:

;NaNvalue: data value in DATA to be replaced with IEEE NaN in the output
; file.
...

So it seamlessly integrates help (and calling syntax, and keyword lists, and completion, and...) from all external libraries too, and even from your own routine collection! This is the reason I have started to use well formed documentation headers again (also because [C-c C-h] inserts one automatically for me, and [C-c Ret] puts the date, my name, etc. for a Modification Entry). Even if you don't have a good doc header, IDLWAVE pulls up the routine declaration.

Never be without routine/syntax/keyword/sysvar/object_method information again.

JD