
Subject: Re: HELP! Annoying IDL glitches...

Posted by [Robert Moss](#) on Thu, 16 May 1996 07:00:00 GMT

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

I can answer a couple of these...

deb wrote:

>

> Here are some questions I'm hoping seasoned IDL users will be able to

> help me out with.

>

snip

>

> 3. Why does XYOUTS, when called with the keyword TEXT_AXES = 3

> rotate text such that it's mirror imaged of what one would expect for

> a regular y-axis label?

TEXT_AXES = 3 writes text as if the Y axis was the horizontal axis, and the X axis was the vertical axis. I'm pretty sure that's why it looks the way it does to you. You can use the ORIENTATION keyword to flip it around if you want, or use a different value for TEXT_AXES. It's all in the XYOUTS help file.

>

snip

>

> 5. Can one open a single (large) graphics window with scroll bars?

> (Not the same as the SLIDE_IMAGE command, but along the lines of the

> scroll window which is incorporated into that)

>

Piece of cake. Suppose you want a 1000x1000 window, with a 512x512 viewable area, with scroll bars. Try this:

```
base = widget_base()
draw = widget_draw( base, xsize = 1000, ysize = 1000, $
  x_scroll_size = 512, y_scroll_size = 512 )
widget_control, base, /realize
widget_control, draw, get_value = win_num
wset, win_num
```

There you have it

> 6. Suppose one opens a datafile which contains an ascii header and then

snip

> work ok. When the binary junk is not stripped out, sometimes a STRPOS

> command finds the tag but frequently IDL chokes; when using a simple readf

> command, the error message says the input line is too long for the input
> buffer,
> even though it really isn't anywhere near 32767 characters long. Why?
> How does
> one get around this?

Well, the I/O buffer size is platform specific. For example, on SunOS machines it defaults to 2048 bytes. From your comments I'd say this is the root of your problem. All you need to do is use the BUFSIZ keyword to OPENR, setting it to an appropriate large value:

```
OPENR, ilun, "foo.bar", BUFSIZ = 32767, /GET_LUN
```

As to your more general problem of finding the ASCII tag before your data, I'm sure there are at least as many ways of doing this as there are people to answer your question. I suggest reading the whole thing in as a byte array (or some sufficiently large portion of it, if reading the whole thing is unreasonable) and using the READS procedure to parse it into whatever bits of whatever type you want.

>
> -Deb Summa
> summa@lanl.gov

Good luck. Feel free to contact me via email if you have any questions.

--
Robert M. Moss, Ph.D. - mossrm@texaco.com

This does not necessarily reflect the opinions of Texaco Inc.
