Subject: Re: Variable stride in array indices and other enhancements
Posted by Jack Saba on Wed, 19 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

"Kenneth P. Bowman" wrote:
>
> In article <ySg03.87161$A6.43176220@news1.teleport.com>, "DBorland"
> <dborland@egi.com> wrote:
>
>> IDL> a[(a[*,2*LINDGEN(3)])[2*LINDGEN(3),*]] = -1
>>
>> When you do this, the values from above are set to -1
:
> I still like
>
> a[0:*:2,0:*:2] = -1
>
> for aesthetic reasons alone.

The format that worked IS aesthetic disgusting. More important, it's unwieldy, and it's more difficult than necessary to figure out the correct syntax for any individual case.

Given the post by Richard French noting that IDL 5.3 is due out in October,
maybe it's time to send RSI a wish list. Here are a few items
I've thought of. At least some have been mentioned in the newsgroup before.

1. direct access to the routines shade_surf uses to compute the shading of a surface and that shade_surf and surface use for hidden-line removal. I'm hoping this will allow direct construction of combined images such as those on Struan Gray's web page (http://www.sljus.lu.se/stm/IDL/Surf_Tips/), eliminating the need to go through TVRD, which limits resolution. And maybe IDL will fix the problem that causes the wire-frame lines to come out below the surface in the Z buffer. Or maybe RSI can come up with another way to accomplish this.

2. /all keyword for free_lun and wdelete.

3. LONG integer indices in ALL idl standard functions and procedures to avoid problems with loops failing because of integer overflow of the loop index.

4. automatic REFORM where needed so you don't get those annoying messages about arrays being the wrong shape when the only problem is a degenerate dimension on an array. This is sometimes the user's

doing, but more frequently it's the result of some IDL function that returns an array even if it has only one element in it.

5. the ablilty to be able to use a=g where a and g are anonymous structures with exactly the same fields. There is a procedure in the idlastro library that can do this, but it would be better if the functionality were built in.

6. a way to force structures to pack densely, without padding, so they can be used for I/O in packed data files.

7. the ability to read ASCII data files that were written by IDL without having to use formats. Two specific problems:
   a. when you PRINTF a structure, the braces are included in the output. If you try to READF the structure, you get an input conversion error.
   b. PRINTF will under some circumstances write data without leaving spaces between numbers if the values are <0. The resulting file cannot be read with READF unless you use a FORMAT.

8. And Ken's request: the ability to address arrays with a stride, e.g., a[0:5:2,3:*:3].