
Subject: Re: Variable stride in array indices and other enhancements

Posted by [davidf](#) on Wed, 19 May 1999 07:00:00 GMT

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

Kenneth P. Bowman (bowman@null.tamu.edu) writes:

- > RSI has greatly expanded IDL in many directions (ENVI, RiverTools, etc.),
- > but it is still difficult to get a correct rectangular border around a
- > simple cylindrical-equidistant global map. Every one I draw has strange
- > artifacts.

Well, really. How much money would it take to get you to work on rectangular borders instead of something neat in today's programming marketplace?

I'm not offering excuses. I'm just saying I wouldn't want the job either. It's the difference between hitting balls with the 15 year old hot-shot or his 10 year old brother. It's not that it isn't important. It's just that it's not as much fun. (And, anyway, his mother has more patience.)

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

Subject: Re: Variable stride in array indices and other enhancements

Posted by [bowman](#) on Wed, 19 May 1999 07:00:00 GMT

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

In article <3742C069.AA4EFE3C@icesat1.gsfc.nasa.gov>, Jack Saba <jack@icesat1.gsfc.nasa.gov> wrote:

- > 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.

I submitted the 'stride' request to the IDL comment form at

<http://www.rsinc.com/contactus/feedback.cfm>

Anyone care to second it?

I also submitted a request to have MAP_IMAGE handle 24-bit images (instead of having to call it 3 times, once for each image plane).

I also agree with the sentiment expressed by Struan Gray in another thread:

- > That said, if history is anything to go by, 5.3 will leave intact
- > the bugs identified by Moses back in version 0.1b5, while presenting a
- > radical new way to 'simplify' programming on Windows 3.1 (only) which
- > ensures nice long coffee breaks for any user daft enough to plot
- > arrays with more than about ten elements.

RSI has greatly expanded IDL in many directions (ENVI, RiverTools, etc.), but it is still difficult to get a correct rectangular border around a simple cylindrical-equidistant global map. Every one I draw has strange artifacts.

Another thing that drives me nuts is differences between the Hershey and Postscript font character ordering. Everytime I want to convert a plot from X to PS, I have to insert special code to handle any special characters.

Another of my pet peeves - no real 24-bit support for the Postscript device.

All that said, many things have been improved in the internals over the years. I love /NAN in TOTAL, etc. I would like to see continued efforts by RSI to make every detail in IDL right. Keep reporting those bugs (I mean features)!

Ken Bowman

Subject: Re: Variable stride in array indices and other enhancements
Posted by [Craig Markwardt](#) on Wed, 19 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jack Saba <jack@icesat1.gsfc.nasa.gov> writes:

- >
- > 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.
- >
- > ... eight requests deleted ...

I agree with what's been said so far. How about a few more requests, in increasing order of drastic-ness. These are still mostly

language-completeness issues, not feature requests.

1. Procedures and functions which do not accept any keywords, should at least accept `_EXTRA=<undefined>`. At the moment IDL crashes when this happens.
2. `CONGRID()` returns interpolates which are off by half a pixel. I argue this is incorrect, and it should be fixed.
3. An interrogation routine to determine whether a file is open or not. `HELP, /FILES` is not enough
4. Additional keyword to `MAKE_ARRAY`, `DBLARR`, etc, which enforces strict dimensions. The final dimension should not be dropped!
5. Ability to query current line number and filename, for reporting error messages. The `HELP` procedure is sometimes overkill.
6. Ability to implicitly index an array relative to its last element:
`a(0:* -2) = 1` ; sets all but last two elements to 1

7. The `WHERE()` function

* should return a `NULL` value upon request, instead of `-1`.

`a(NULL) = 1` ; would have no effect

* should have a `COMPLEMENT` keyword, to return indices of array elements that fail the test.

`A_IS_1 = WHERE(A EQ 1, COMPLEMENT=A_NOT_1)`

This can save performing `WHERE()` twice on large arrays.

8. Ability to access strings as arrays, rather than with `STRMID()` and `STRPUT`.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Variable stride in array indices and other enhancements

Posted by [davidf](#) on Wed, 19 May 1999 07:00:00 GMT

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

Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:

- > Perhaps we could design a specification for a function to dice an
- > existing array in memory which accepted similar keywords.

Now here is a useful and sensible suggestion. I nominate Phil Aldis to knock it out for us. It will provide some distraction from that &%*#@ \$ table widget.

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

Subject: Re: Variable stride in array indices and other enhancements

Posted by [Liam Gumley](#) on Wed, 19 May 1999 07:00:00 GMT

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

Jack Saba wrote:

- > I agree. The format that worked is not only REALLY ugly, it's unwieldy.
- > It's also more difficult than necessary to figure out the correct syntax
- > for any individual case.

No argument from me.

In many cases, I suspect people just want to do what Ken asked: extract every nth element along each dimension. Perhaps we could come up with a specification for an array dicing routine.

The closest paradigm in IDL that I'm aware of is the netCDF routine NCDF_VARGET (see the Scientific Data Formats documentation). It is a general purpose routine for reading multi-dimensional arrays from netCDF files. It also recognizes that users might want to read subsets of an array, including skipping elements along each dimension. The following optional keywords to NCDF_VARGET are allowed in IDL 5.2:

COUNT

An optional vector containing the counts to be used in reading Value.

COUNT is a 1-based vector with an element for each dimension of the data

to be written. The default matches the size of the variable so that all data is written out.

OFFSET

An optional vector containing the starting position for the read. The default start position is [0, 0, ...].

STRIDE

An optional vector containing the strides, or sampling intervals, between accessed values of the netCDF variable. The default stride vector is that for a contiguous read, [1, 1, ...].

Perhaps we could design a specification for a function to dice an existing array in memory which accepted similar keywords.

Cheers,
Liam.

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>

Subject: Re: Variable stride in array indices and other enhancements

Posted by [davidf](#) on Wed, 19 May 1999 07:00:00 GMT

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

Jack Saba (jack@icesat1.gsfc.nasa.gov) writes:

> 2. /all keyword for free_lun and wdelete.

Allow me to help in this regard.

For freeing up all open file units (whether opened with GET_LUN or hard-coded):

Close, /All

For deleting all graphics windows (including those contained in widget programs), I offer this program:

```
PRO WDeleteAll
While !D.Window NE -1 DO WDelete, !D.Window
Widget_Control, /Reset
END
```

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

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 ability 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].

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](#)

I agree. The format that worked is not only REALLY ugly, it's unwieldy. It's also more difficult than necessary to figure out the correct syntax for any individual case.

Given the post by Richard French noting 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:

"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

>> IDL> print,a

```
>>      -1      1      -1      3      -1      5
>>      6      7      8      9      10     11
>>     -1     13     -1     15     -1     17
>>     18     19     20     21     22     23
>>     -1     25     -1     27     -1     29
>>     30     31     32     33     34     35
```

>

> This only works because the original array was created with LINDGEN. It

> won't work in the general case.

>

> I still like

>

> a[0:*:2,0:*:2] = -1

>

> for aesthetic reasons alone.

>

> Ken