

---

Subject: Re: Incremental variable names?

Posted by [mmeron](#) on Mon, 02 Feb 2004 04:51:32 GMT

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

---

In article <BC430FE2.18E9C%greenberg@ucdavis.edu>, Jonathan Greenberg <greenberg@ucdavis.edu> writes:

> Have a question about variable names. Is there any way to create and then  
> call variable names that have some incremental value in them, e.g.:

>

> t0

> t1

> t2

>

> Etc... I want to be able to first create and assign a value to tn where n =

> 0 to whatever, and then be able to subsequently call that variable using a

> loop.

>

Yes, using EXECUTE. You can find some examples in my library (MIDL,  
on the RSI users contribution page). Look in the routines HOW\_MANY  
and OUTPUT, for example.

Mati Meron | "When you argue with a fool,  
meron@cars.uchicago.edu | chances are he is doing just the same"

---

---

Subject: Re: Incremental variable names?

Posted by [David Fanning](#) on Mon, 02 Feb 2004 04:55:02 GMT

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

---

Jonathan Greenberg writes:

> Have a question about variable names. Is there any way to create and then  
> call variable names that have some incremental value in them, e.g.:

>

> t0

> t1

> t2

>

> Etc... I want to be able to first create and assign a value to tn where n =

> 0 to whatever, and then be able to subsequently call that variable using a

> loop.

Here is how I do this in one of my programs. In this  
case, the files are named image01.tif, image02.tif...  
image53.tif.

```
FOR j=0, 53 DO BEGIN
```

```
  filename = 'image' + String(j, Format='(I2.2)') + '.tif'
```

```
Write_Tiff, filename, Reverse(Reform(head[*,*],j)),2)
ENDFOR
```

Cheers,

David

--

David W. Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Phone: 970-221-0438, E-mail: david@dfanning.com  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

---

Subject: Re: Incremental variable names?  
Posted by [David Fanning](#) on Mon, 02 Feb 2004 04:57:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning writes:

```
> Here is how I do this in one of my programs. In this
> case, the files are named image01.tif, image02.tif...
> image53.tif.
>
> FOR j=0, 53 DO BEGIN
>   filename = 'image' + String(j, Format='(I2.2)') + '.tif'
>   Write_Tiff, filename, Reverse(Reform(head[*,*],j)),2)
> ENDFOR
```

Whoops! Sorry, I'm still thinking about Janet Jackson and the Super Bowl.

I'll try to answer the real question next time. :-(

Cheers,

David

--

David W. Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Phone: 970-221-0438, E-mail: david@dfanning.com  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>  
Toll-Free IDL Book Orders: 1-888-461-0155

---

---

Subject: Re: Incremental variable names?

Posted by [btt](#) on Mon, 02 Feb 2004 13:57:53 GMT

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

---

Jonathan Greenberg wrote:

> Have a question about variable names. Is there any way to create and then  
> call variable names that have some incremental value in them, e.g.:  
>  
> t0  
> t1  
> t2  
>  
> Etc... I want to be able to first create and assign a value to tn where n =  
> 0 to whatever, and then be able to subsequently call that variable using a  
> loop.

Hello,

I'm a bit confused about why you to name the variable incrementally if what you really want to access them incrementally. It seems to me that you will not be able to decode the increment of the variable with ease after you have created it. I would seriously consider using a linked list (<http://www.dfanning.com/documents/programs.html#LINKEDLIST>) or make a variable object that can be stored in a container (which is just a linked list itself.) Then you simply access the variable in the order they have been stored.

I was introduced to the latter method when Martin Schultz (where is he, anyway?) first built a generic object system. I prefer the Martin's method now, as I can do some spiffy things with the variables without cluttering up my code.

Cheers,  
Ben

---

---

Subject: Re: Incremental variable names?

Posted by [MKatz843](#) on Mon, 02 Feb 2004 16:52:39 GMT

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

---

Jonathan, I apologize if this solution is way too obvious and you may have something else in mind. Have you considered an array?

```
t = fltarr(10)
t(0)
t(1)
t(2)
```

If the values all need to be of different types, then pointers come to the rescue.

```
t = ptrarr(10)
t(0) = ptr_new(0.5)
t(1) = ptr_new('hello')
t(2) = ptr_new(dblarr(100,400))
print, *t(0)
print, *t(1)
    and so forth
```

Plus you can extend the length at will  
t = [t, ptr\_new('extension works')]  
or remove elements, retaining only some.  
t = t([1, 3, 4])

Is there a reason why this wouldn't work for you?

M. Katz.

Jonathan Greenberg <greenberg@ucdavis.edu> wrote in message  
news:<BC430FE2.18E9C%greenberg@ucdavis.edu>...

```
> Have a question about variable names. Is there any way to create and then
> call variable names that have some incremental value in them, e.g.:
>
> t0
> t1
> t2
>
> Etc... I want to be able to first create and assign a value to tn where n =
> 0 to whatever, and then be able to subsequently call that variable using a
> loop.
>
> --j
```

---

Subject: Re: Incremental variable names?

Posted by [condor](#) on Mon, 02 Feb 2004 21:45:06 GMT

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

---

Jonathan Greenberg <greenberg@ucdavis.edu> wrote in message  
news:<BC430FE2.18E9C%greenberg@ucdavis.edu>...

```
> Have a question about variable names. Is there any way to create and then
> call variable names that have some incremental value in them, e.g.:
>
> t0
> t1
> t2
>
> Etc... I want to be able to first create and assign a value to tn where n =
> 0 to whatever, and then be able to subsequently call that variable using a
```

> loop.  
>

OK, now tell us what you REALLY want to do. ;-)

As it is written up there, there doesn't exactly seem to be a reason not to use an array for your variables -- instead of `tn` you write `t[n]` and that's pretty much it.

I assume that there's a reason for you not to do things that way.

Now while there is certainly the possibility to use EXECUTE, I'd be surprised if there weren't some cleverer way to do whatever you're trying to do and I'm sure a lot of people would be happy to tell you about that way if we had a little more information at hand ...

---

Subject: Re: Incremental variable names?  
Posted by [Jonathan Greenberg](#) on Tue, 03 Feb 2004 05:46:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I should get in the practice of not making "examples" of what I want to do and just come out and say it:

OK, from a previous email we were discussing extracting "rings" of data from an image to calculate a semivariance for each pixel in the image. The previous suggestions helped a lot in optimizing how to extract the data from the array (use `where` + an array of index values of the ring locations offset from the center pixel).

I'm trying to optimize this algorithm for use with large images. ENVI/IDL work well if you extract a single line of an image at a time (line = y axis, by the way). For a circular mask, however, I'd need to extract at least as many lines as the maximum circular diameter (e.g. If my circle is 3 pixels in radius, I need to pull out 7 lines of data to figure out the values along the ring).

It occurred to me (and tell me if I'm wrong), that while I CAN extract the line I'm working on +/- 3 lines from the image each time, this seems like an awful waste of I/O, when in reality the next line will use all but one of the previous extracted rows (e.g. I just need to read the next line in, and delete the current top line to free up memory). Creating arrays of the lines would require rewriting the arrays each time (e.g. `Arraynew = (arrayold(2:7),newline)` -- I know this isn't idl code, but you get the idea) which seems slower than just having 7 named variables (`line1,line2,...line7`), doing my calculation, then deleting `line1`, and adding `line8` and repeating.

No? Is there a better way of "preserving" the data I've already read from the HD than using incremental variable names?

--j

On 2/2/04 1:45 PM, in article  
df160b8f.0402021345.25b58a8@posting.google.com, "Big Bird"  
<condor@biosys.net> wrote:

```
> Jonathan Greenberg <greenberg@ucdavis.edu> wrote in message
> news:<BC430FE2.18E9C%greenberg@ucdavis.edu>...
>> Have a question about variable names. Is there any way to create and then
>> call variable names that have some incremental value in them, e.g.:
>>
>> t0
>> t1
>> t2
>>
>> Etc... I want to be able to first create and assign a value to tn where n =
>> 0 to whatever, and then be able to subsequently call that variable using a
>> loop.
>>
>
>
> OK, now tell us what you REALLY want to do. ;-)
>
> As it is written up there, there doesn't exactly seem to be a reason
> not to use an array for your variables -- instead of tn you write t[n]
> and that's pretty much it.
>
> I assume that there's a reason for you not to do things that way.
>
> Now while there is certainly the possibility to use EXECUTE, I'd be
> surprised if there weren't some cleverer way to do whatever you're
> trying to do and I'm sure a lot of people would be happy to tell you
> about that way if we had a little more information at hand ...
```

---

---

Subject: Re: Incremental variable names?  
Posted by [Chris Lee](#) on Wed, 04 Feb 2004 09:34:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <BC4475B7.18FD5%greenberg@ucdavis.edu>, "Jonathan Greenberg"  
<greenberg@ucdavis.edu> wrote:

> I'm trying to optimize this algorithm for use with large images.  
 > ENVI/IDL work well if you extract a single line of an image at a time  
 > (line = y axis, by the way). For a circular mask, however, I'd need to  
 > extract at least as many lines as the maximum circular diameter (e.g. If  
 > my circle is 3 pixels in radius, I need to pull out 7 lines of data to  
 > figure out the values along the ring).  
 > It occurred to me (and tell me if I'm wrong), that while I CAN extract  
 > the line I'm working on +/- 3 lines from the image each time, this seems  
 > like an awful waste of I/O, when in reality the next line will use all  
 > but one of the previous extracted rows (e.g. I just need to read the  
 > next line in, and delete the current top line to free up memory).  
 > Creating arrays of the lines would require rewriting the arrays each  
 > time (e.g. Arraynew = (arrayold(2:7),newline) -- I know this isn't idl  
 > code, but you get the idea) which seems slower than just having 7 named  
 > variables (line1,line2,...line7), doing my calculation, then deleting  
 > line1, and adding line8 and repeating.  
 > No? Is there a better way of "preserving" the data I've already read  
 > from the HD than using incremental variable names? --j  
 > On 2/2/04 1:45 PM, in article

If I understand your method correctly.....Create a 2D array with the  
 dimension of (radius, line length). Each time you read in a new line..

```

array=fltarr(radius, line_length)
input=fltarr(line_length)
openr, file, "filename", /get_lun
counter=0L
cmax=2*radius+1
while(not eof(file)) do begin
  readu,file, input
  array[counter mod cmax,0]=input
  array=shift(array, 1,0) ;;;This could be wrong, try -1,0...
  result=do_stuff(array)
  counter=counter+1
endwhile

```

free\_lun, file

; I think that would work..

Of course, if I understand this whole ring thing, your doing this with  
 every point on your image? If you are, the function DO\_STUFF above (your  
 semi-variance code) had better be vectorised along the line\_length.

Chris.

---