Subject: Reset Pointer at End of For Loop Issue
Posted by Lisa08 on Tue, 30 Apr 2013 20:37:12 GMT
View Forum Message <> Reply to Message

Hi,

I have run into something in IDL that has me a bit confused. I am trying to run this script to make an interval of day_of_year values (bounded by lowlim and uplim) that IDL will later use to find data that corresponds to the days specified in that interval. The lowlim and uplim is supposed to change during each iteration. If I set the time_step as 30 so that it sets the interval limits as 0 to 30, 30 to 60, etc, it works fine. However, when I try to use the specified limits for each month, low_daylim and up_daylim, it only makes it to day 31 and then spits out an error that says:

% Expression must be a scalar in this context: DIDX.

I can't set the time_step as 30 since I am trying to do a monthly analysis and not all months have 30 days. Consequently, I have to use the month limits that I specify in the beginning of the routine. If this is an issue with the pointer being reset at the end of the for loop I don't understand why it would work in one case and not in the other. Can somebody explain to me where I have gone wrong? I just don't see it.

Here is the routine that spits out the error:

```
PRO test_loop

;First day of each month
low_daylim=[1,32,61,92,122,153,183,214,245,275,306,336]

;Last day of each month
upper_daylim=[31,60,91,121,152,182,213,244,274,305,335,366]

;First day to begin loop
day_begin=1

;Last day to begin loop
day_end=366

;Number of days in interval
time_step=30


FOR didx=day_begin,day_end DO BEGIN
```

```
limits=where(low_daylim EQ didx)
lowlim=didx-1
uplim=upper_daylim[limits]


; lowlim=didx-1
; uplim=didx+(time_step-1)

 print, 'lowlim'
 print, lowlim

 print, 'uplim'
 print, uplim


 ;Set didx as uplim before next iteration
 didx=uplim
 print, 'didx'
 print, didx


ENDFOR



STOP
END
```

The other one (where the time_step is set as 30) works when you comment out this part:

```
limits=where(low_daylim EQ didx)
lowlim=didx-1
uplim=upper_daylim[limits]
```

and uncomment the two lines below this.

If you need any more information from me or I haven't made something very clear please let me know. I tried using a ptr_free before setting the didx to uplim at the end of the for loop but I really didn't expect it to work. Any suggestions would be greatly appreciated as I have been stuck for a while.

Subject: Re: Reset Pointer at End of For Loop Issue
Posted by Dick Jackson on Tue, 30 Apr 2013 21:06:07 GMT

Hi lrod09,

It's good that you included just enough code for us to see the problem.

At the point where it stops, it may be hard to tell, but 'didx' is not a scalar (single value) (and the error message was saying so), it is an array. It looks like it's just a scalar when printed, and if you're looking in the workbench Variables pane it may look like a scalar, until you add the 'Type' column to the Variables display. Or, if you do this you see it's a one-element array:

```
IDL> help,didx
DIDX            INT       = Array[1]
```

The error appears to be on the Print statement, but the problem is actually on the ENDFOR, where IDL is trying to increment 'didx', and it is no longer a scalar as it had started out, and which is required for a FOR loop variable.

You can figure out how 'didx' became an array, originally from the Where() statement which, even when one value matches, it returns an array. If you're sure it will only match one (as you are here) you can change the line to

```
limits = (where(low_daylim EQ didx))[0] ; Take first element as scalar
```

... and 'limits' will be a scalar, then 'uplim' will become a scalar, etc.

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada --- +1-250-220-6117
dick@d-jackson.com --- http://www.d-jackson.com

lrod09 wrote:
> Hi,
>
> I have run into something in IDL that has me a bit confused. I am trying to run this script to make an interval of day_of_year values (bounded by lowlim and uplim) that IDL will later use to find data that corresponds to the days specified in that interval. The lowlim and uplim is supposed to change during each iteration. If I set the time_step as 30 so that it sets the interval limits as 0 to 30, 30 to 60, etc, it works fine. However, when I try to use the specified limits for each month, low_daylim and up_daylim, it only makes it to day 31 and then spits out an error that says:
>
> % Expression must be a scalar in this context: DIDX.
>
>
> I can't set the time_step as 30 since I am trying to do a monthly analysis and not all months

have 30 days. Consequently, I have to use the month limits that I specify in the beginning of the routine. If this is an issue with the pointer being reset at the end of the for loop I don't understand why it would work in one case and not in the other. Can somebody explain to me where I have gone wrong? I just don't see it.

```
>
>
>
>  Here is the routine that spits out the error:
>
>
>  PRO test_loop
>
>  ;First day of each month
>  low_daylim=[1,32,61,92,122,153,183,214,245,275,306,336]
>
>  ;Last day of each month
>  upper_daylim=[31,60,91,121,152,182,213,244,274,305,335,366]
>
>  ;First day to begin loop
>  day_begin=1
>
>  ;Last day to begin loop
>  day_end=366
>
>  ;Number of days in interval
>  time_step=30
>
>
>  FOR didx=day_begin,day_end DO BEGIN
>
>
>  limits=where(low_daylim EQ didx)
>  lowlim=didx-1
>  uplim=upper_daylim[limits]
>
>
>  ; lowlim=didx-1
>  ; uplim=didx+(time_step-1)
>
>    print, 'lowlim'
>    print, lowlim
>
>    print, 'uplim'
>    print, uplim
>
>
>    ;Set didx as uplim before next iteration
>    didx=uplim
```

> print, 'didx'
> print, didx
>
>
> ENDFOR
>
>
>
> STOP
> END
>
>
>
> The other one (where the time_step is set as 30) works when you comment out this part:
>
> limits=where(low_daylim EQ didx)
> lowlim=didx-1
> uplim=upper_daylim[limits]
>
> and uncomment the two lines below this.
>
>
> If you need any more information from me or I haven't made something very clear please let me know. I tried using a ptr_free before setting the didx to uplim at the end of the for loop but I really didn't expect it to work. Any suggestions would be greatly appreciated as I have been stuck for a while.
>
>

--

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada
www.d-jackson.com

Subject: Re: Reset Pointer at End of For Loop Issue
Posted by lroubertrod on Tue, 30 Apr 2013 21:47:31 GMT
View Forum Message <> Reply to Message

On Tuesday, April 30, 2013 3:37:12 PM UTC-5, lrod09 wrote:
> Hi,
>
>

>
> I have run into something in IDL that has me a bit confused. I am trying to run this script to make an interval of day_of_year values (bounded by lowlim and uplim) that IDL will later use to find data that corresponds to the days specified in that interval. The lowlim and uplim is supposed to change during each iteration. If I set the time_step as 30 so that it sets the interval limits as 0 to 30, 30 to 60, etc, it works fine. However, when I try to use the specified limits for each month, low_daylim and up_daylim, it only makes it to day 31 and then spits out an error that says:
>
>
>
> % Expression must be a scalar in this context: DIDX.
>
>
>
>
>
> I can't set the time_step as 30 since I am trying to do a monthly analysis and not all months have 30 days. Consequently, I have to use the month limits that I specify in the beginning of the routine. If this is an issue with the pointer being reset at the end of the for loop I don't understand why it would work in one case and not in the other. Can somebody explain to me where I have gone wrong? I just don't see it.
>
>
>
>
>
>
>
> Here is the routine that spits out the error:
>
>
>
>
>
> PRO test_loop
>
>
>
> ;First day of each month
>
> low_daylim=[1,32,61,92,122,153,183,214,245,275,306,336]
>
>
>
> ;Last day of each month
>
> upper_daylim=[31,60,91,121,152,182,213,244,274,305,335,366]
>

```
>
>
>  ;First day to begin loop
>
>  day_begin=1
>
>
>
>  ;Last day to begin loop
>
>  day_end=366
>
>
>
>  ;Number of days in interval
>
>  time_step=30
>
>
>
>
>
>  FOR didx=day_begin,day_end DO BEGIN
>
>
>
>
>
>  limits=where(low_daylim EQ didx)
>
>  lowlim=didx-1
>
>  uplim=upper_daylim[limits]
>
>
>
>
>
>  ; lowlim=didx-1
>
>  ; uplim=didx+(time_step-1)
>
>
>
>   print, 'lowlim'
>
>   print, lowlim
>
```

```
>
>
>   print, 'uplim'
>
>   print, uplim
>
>
>
>
>
>   ;Set didx as uplim before next iteration
>
>   didx=uplim
>
>   print, 'didx'
>
>   print, didx
>
>
>
>
>
>  ENDFOR
>
>
>
>
>
>
>
>
>
>  STOP
>
>  END
>
>
>
>
>
>
>
>
>  The other one (where the time_step is set as 30) works when you comment out this part:
>
>
>
>  limits=where(low_daylim EQ didx)
>
>  lowlim=didx-1
>
```

> uplim=upper_daylim[limits]
>
>
>
> and uncomment the two lines below this.
>
>
>
>
>
>
> If you need any more information from me or I haven't made something very clear please let me know. I tried using a ptr_free before setting the didx to uplim at the end of the for loop but I really didn't expect it to work. Any suggestions would be greatly appreciated as I have been stuck for a while.

Hi there,

Thank you so much for your detailed response. I definitely see the issue now. I definitely did assume that didx was a scalar a scalar the whole time since it always was a single value. I'll make sure to keep an eye on things like that in the future. Thanks so much for your help.

Thanks again,

lrod09

Subject: Re: Reset Pointer at End of For Loop Issue
Posted by David Fanning on Tue, 30 Apr 2013 22:00:39 GMT
View Forum Message <> Reply to Message

lroubertrod@gmail.com writes:

> Thank you so much for your detailed response. I definitely see the issue now. I definitely did assume that didx was a scalar a scalar the whole time since it always was a single value. I'll make sure to keep an eye on things like that in the future. Thanks so much for your help.

Well, if you are like the rest of us, it will bite you at least two more
times before you REALLY learn keep it in mind. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")