

---

Subject: Re: Attempt to subscript FILNAM with I is out of range....  
Posted by [R.Bauer](#) on Fri, 16 Aug 2002 09:08:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Kolbjorn Bekkelund wrote:

> Hi again, David.  
>  
> As I said; When running FOR statements in batch mode the IDL help said  
> that I had to connect all lines after BEGIN until ENDFOR to one line. I  
> did this by using & \$ after each line in the original FOR statement.  
>  
> I ended up with this:

Dear Kolbjorn

I was believing that's in idl5.5 the "& \$" isn't any more necessary.  
Am I wrong? Or did you use idl 5.4 ?

One more comment to your code. you should use [] for arrays  
and () for functions. itim(i) is an array.  
If you always use only () you can get in a conflict if sometimes  
a variable is named like a function.

Labels as Label1: are very bad program style in my eyes and the code  
could be very hard to understand.

Did you have thought about an idl runtime version of your program,  
Bevause it looks like a procedure with one input parameter but this  
isn't used it's overwritten by the findfile function.

I believe more you have an older version because findfile is replaced  
by file\_search in idl5.5.

These steps are required

```
.compile batch_calc_quality_dual_online_auto  
resolve_all  
save,file='batch_calc_quality_dual_online_auto.sav',/routine s,/compress
```

Now you can execute in a cron job the runtime routine

```
idl -rt="/batch_calc_quality_dual_online_auto.sav"
```

If you do so you can work together with your development license while  
this job is running.

regards

Reimar

```
>
> PRO BATCH_CALC_QUALITY_DUAL_ONLINE_AUTO, filnam
> device, decomposed = 0
> loadct, 23
>
>   fil_dir = "
>   fil_dir = '/data/ozon/dialdata/ozon-online/*'
>   filnam = findfile(fil_dir)
>   iff = N_ELEMENTS(filnam)
>
> dummy = "
>   idat = intarr(3) & idat0 = 0 & dat = fltarr(2) & idat1 = lonarr(4)
>   ref308 = fltarr(iff) & ref353 = fltarr(iff) & itim = fltarr(iff)
>   back308 = fltarr(iff) & back353 = fltarr(iff) & dz = fltarr(iff)
>   hr = 0 & min = 0 & stim = 0. & etim = 0.
>   hr='00'
>   min='00'
>   stim = hr + min/60.
>   hr='24'
>   min='00'
>   etim = hr + min/60.
>   IF etim LT stim THEN etim = etim + 24.
>
> FOR i = 0, iff-1 DO BEGIN & $
>   openr, resf, filnam(i), /get_lun & $
>   filnavn=STRMID(filnam(i), 32,5) & $
>   FOR j = 0, 5 DO readf, resf, dummy & $
>   readf,resf,format = '(a36,i2,1x,i2,1x,i2)',dummy,idat & $
>   itim(i) = idat(0) + (idat(1) + idat(2)/60.)/60. & $
>   IF itim(i) LT itim(0) THEN itim(i) = itim(i) + 24. & $
>   IF itim(i) LT stim OR itim(i) GT etim THEN GOTO, label1 & $
>
>   readf, resf, dummy & $
>   readf, resf, format = '(a24,i6)', dummy, ishot & $
>   FOR j = 0, 5 DO readf, resf, dummy & $
>   iback = 0 & $
>   WHILE NOT eof(resf) DO BEGIN & $
>     readf,resf,idat0,dat,idat1 & $
>     IF dat(1) EQ 19.993 THEN BEGIN & $
>       dz(i) = dat(0)/1000. & $
```

```

>     ref308(i) = idat1(0)/dz(i) & $
>     ref353(i) = idat1(2)/dz(i) & $
> ENDIF & $
> IF dat(1) GE 85. AND dat(1) LT 87. THEN BEGIN & $
>     back308(i) = back308(i) + idat1(0)/dz(i) & $
>     back353(i) = back353(i) + idat1(2)/dz(i) & $
>     iback = iback + 1 & $
> ENDIF & $
> ENDWHILE & $
> back308(i) = (back308(i)/iback)/ishot & $
> back353(i) = (back353(i)/iback)/ishot & $
>
> ref308(i) = (ref308(i)/ishot)-back308(i) & $
> ref353(i) = (ref353(i)/ishot)-back353(i) & $
> Label1: & $
> free_lun, resf & $
> ENDFOR          ; i
>
>
> Now I can run 'idl batch_calc_quality_dual_online_auto.pro' and get my
> plots via a cron-job in Linux.
>
> Thanks for bearing over with me :-)
>
> Kolbjorn
>
> On Thu, 15 Aug 2002 18:44:34 +0000, David Fanning wrote:
>
>> Kolbjorn Bekkelund (kobe@rocketrange.no) writes:
>>
>>> I'm following up my own last post since I found that one cannot use FOR
>>> statements in batch mode like in normal .run. You have to place all the
>>> FOR stuff (until the ENDFOR) on one line using $ and &. This because
>>> the batch mode use of IDL is looking for an END statement on each line
>>> after the BEGIN. If it doesn't find it you get a syntax error.
>>>
>>> BUT:
>>> How do I split the below FOR ..... ENDFOR using $ and &, and still make
>>> it work ???
>>
>> Kolbjorn,
>>
>> You *really* don't want to do this!
>>
>> You have written the code as a procedure. Why not just use it as a
>> procedure? (Put the name of the procedure in your batch file.) What am I
>> missing!?
>>

```

>> David

>

>

---