

---

Subject: Re: WHERE approaching Flatland  
Posted by [David Fanning](#) on Tue, 18 Feb 2003 20:15:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sean Raffuse (sean@me.wustl.edu) writes:

```
> Is there are 2-d version of where?
>
> I would like to do this:
>
> bad = where(Array[*,*,0] GT Array[*,*,5])
>
> Array[bad,1:5] = Array[bad,0]
>
> I know that doesn't work, but after having spent some time with IDL, I'm
> allergic to loops.
```

I think maybe you want to re-read that thread from a week or so ago, entitled "Interesting Where Function Gotcha".

Cheers,

David

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

---

Subject: Re: WHERE approaching Flatland  
Posted by [Wonko\[3\]](#) on Tue, 18 Feb 2003 21:48:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

sean@me.wustl.edu (Sean Raffuse) wrote:

```
> Is there are 2-d version of where?
>
> I would like to do this:
>
> bad = where(Array[*,*,0] GT Array[*,*,5])
>
> Array[bad,1:5] = Array[bad,0]
```

I do these things like this:

```
xy = (size(Array,/dim))[0] * (size(Array,/dim))[1]
for i = 1, 5 do Array[ bad + i*xy ] = Array[bad]
```

> I know that doesn't work, but after having spent some time with IDL,  
> I'm allergic to loops.

Ummm... this should work:

```
n = n_elements( bad )
Array[ rebin(bad,n,5) + transpose(rebin((lindgen(5)+1)*xy,5,n)) ] =
rebin( array[bad], n, 5 )
```

Alex

--

Alex Schuster    Wonko@wonkology.org    PGP Key available  
alex@pet.mpin-koeln.mpg.de

---

Subject: Re: WHERE approaching Flatland  
Posted by [Haje Korth](#) on Wed, 19 Feb 2003 12:50:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I always use the function WHERETOMULTI below. (Sorry about the line spacing)

Haje

```
***** Beginning of wheretomulti.pro
*****
```

PRO WhereToMulti, Array, Indices, Col, Row, Frame

```
;+
```

```
; NAME: wheretomulti.pro
```

```
;
```

```
; FUNCTION: Convert WHERE output to 2d or 3d indices
```

```
;
```

```
; USAGE: WhereToMulti, Array, Indices, Col, Row, Frame
```

```
;
```

```
; INPUT ARGUMENTS:
```

```
; Array: the array that was WHERE'd
; Indices: the indices returned by WHERE
;
; OUTPUT ARGUMENTS:
; Col: Indices to first dimension.
; Row: Indices to second dimension.
; Frame: Indices to third dimension. Returned only for 3-d array.
;
; OPTIONAL ARGUMENTS:
;
; KEYWORDS:
;
; REQUIRED MODULES:
;
; SIDE EFFECTS:
;
; ERROR HANDLING:
; If Array is not a vector or matrix, all return values are set to zero
; and a message is written to the screen.
;
; NOTES:
;
; HISTORY:
; 1998 Sept 15 J.L.Saba Developed based on code from David Fanning's
```

; web site.

;

;- End of  
prologue -----

s = SIZE ( Array )

NCol = s[1]

Col = Indices MOD NCol

IF s[0] EQ 2 THEN BEGIN ; 2-d array

Row = Indices / NCol

ENDIF ELSE IF s[0] EQ 3 THEN BEGIN ; 3-d array

NRow = s(2)

Row = ( Indices / NCol ) MOD NRow

Frame = Indices / ( NRow \* NCol )

ENDIF ELSE BEGIN ; neither 2d or 3d

Col = 0

Row = 0

Frame = 0

PRINT, 'WhereToMulti called with bad input. Array not a vector or matrix.'

HELP, Array

ENDELSE

RETURN

END

\*\*\*\*\* End of wheretomulti.pro  
\*\*\*\*\*

--

Dr. Haje Korth  
Space Physics Group  
The Johns Hopkins University  
Applied Physics Laboratory  
MS MP3-E128  
11100 Johns Hopkins Road  
Laurel, MD 20723-6099  
USA  
Phone: 240-228-4033 (Washington), 443-778-4033 (Baltimore)  
Fax: 240-228-0386 (Washington), 443-778-0386 (Baltimore)

"Sean Raffuse" <sean@me.wustl.edu> wrote in message  
news:b2u1d0\$1nc\$1@newsreader.wustl.edu...

> Hello,  
>  
> Is there are 2-d version of where?  
>  
> I would like to do this:  
>  
> bad = where(Array[:,0] GT Array[:,5])  
>  
> Array[bad,1:5] = Array[bad,0]  
>  
> I know that doesn't work, but after having spent some time with IDL, I'm  
> allergic to loops.  
>  
> Thanks in advance,  
>  
> Sean  
>  
>  
>

---