
Subject: Re: The continuing saga of WHERE and 2D
Posted by [marc schellens\[1\]](#) on Fri, 28 Feb 2003 06:43:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sean Raffuse wrote:

```
> "David Fanning" <david@dfanning.com> wrote in message
> news:MPG.18c829481a588402989b05@news.frii.com...
>
>> Sean Raffuse (sean@me.wustl.edu) writes:
>>
>>
>>> Ok, so I've found the nifty little WhereToMulti program that converts 1D
>>> where() results back to the original 2d or 3d indices. My question, and
>>>
> I
>
>>> am sensing that it is a dumb one, is how can I use these returned
>>>
> indices
>
>>> properly?
>>>
>>> Example:
>>>
>>> ;the part that works (stolen wholesale from D Fanning's website)
>>> index = WHERE(image EQ test)
>>> s = SIZE(image)
>>> ncol = s(1)
>>> col = index MOD ncol
>>> row = index / ncol
>>>
>>> ;the part I am confused about
>>> image[col, row] = PassedTheTest
>>>
>>> *** Error, too many elements in array. . . and you're ugly. ***
>>>
>> Say what!? I don't think so. :-)
>>
>> Are you sure all your variables are defined here, Sean?
>> Let's see your real code and not this pseudo code.
>>
>>
>
> Ok, you may disagree and same I look fabulous, but I still get the error.
> Here is my code snippet. Let me know if you need more.
>
> HighRedorGreenLand = where(MinArray8[*,*] GT default_Bd8WaterLimit $ ;pixel
> is not water
```

```

> AND ((Tau[*,*,5] GT Tau[*,*,0]) OR (Tau[*,*,4] GT
> Tau[*,*,0]))
> WhereToMulti, MinArray8, HighRedorGreenLand, x_mask, y_mask
> Tau[x_mask,y_mask,*] = default_BelowDetLimit
>
> ENVI> help, MinArray8
> MINARRAY8 INT = Array[3600, 1680]
> ENVI> help, Tau
> TAU FLOAT = Array[3600, 1680, 8]
> ENVI> help, HighRedorGreenLand
> HIGHREDORGREENLAND
> LONG = Array[1407735]
> ENVI> help, x_mask
> X_MASK LONG = Array[1407735]
> ENVI> help, y_mask
> Y_MASK LONG = Array[1407735]
>
>
>

```

Your problem here is that you index only TWO dimensions of a THREE dimensional array with the same number of indices.

If ALL indexing arrays have the same size, the result is a one-dimensional array of that size.

If only one of the dimensions is different, you get an ix1 * ix2 * ix3 array.

1407735^2 (*1) is a huge number, therefore the error.

To prevent it you may use as third index an lonarr(1407735), or copy the slice from the three dim array to two dim and later copy back (or use Liam Gumley's suggestions)

Hope that helps,

marc