
Subject: Re: where, or, and loops. There has to be a better way.

Posted by [wlandsman](#) on Sat, 09 Mar 2013 00:30:05 GMT

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

It is unnecessary to use the WHERE() function. Instead directly make the position array

```
key = (huc EQ new[0]) or (huc EQ new[1]) or (huc EQ new[2])
```

or if you need to loop:

```
pos = (huc eq new[0])
for i=1,N_elements(new)-1 do pos = pos and (huc EQ new[i])
```

It might be possible to improve on this by decimating the array (don't test pixels that have already been flagged), but I believe the coding would be considerably more complicated. --Wayne

On Friday, March 8, 2013 6:47:00 PM UTC-5, AB wrote:

> Please see the below example to demonstrate my problem:

>

>

>

> -----

>

>

>

>

>

> ; This is a large floating point array of river basins

>

> huc = fltarr(4800,4800)

>

>

>

> ; IDs of the basins I want to extract

>

> NEW = [1102,1103,1104,1105]

>

>

>

> ; Find where the basins are

>

> pos = where(huc eq NEW[0] or \$; IS THIS THE ONLY WAY TO FIND THE VALUES IN THE
LARGER ARRAY??

>

> huc eq NEW[1] or \$

>

> huc eq NEW[2] or \$

```
>
> huc eq NEW[3])
>
>
>
> ; Create new array and insert basins
>
> key = intarr(4800,4800)
>
>
>
> key(pos) = 1
>
> -----
>
>
>
> The issue here is, what happens when the 'NEW' variable has 300 values in it? Do I have to
go through 300 iterations of:
>
>
>
> pos = where(huc eq NEW[1] or $
>
> ..... or $
>
> huc eq NEW[299])
>
>
>
>
> I hope someone has found a way around this.
>
>
>
> Thanks for any ideas you can give!!!
>
> Annie
```
