
Subject: Re: Finding a value in a array efficiently
Posted by [Wayne Landsman](#) on Fri, 14 Apr 2006 01:58:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2006.04.13.23.00.45.990566@as.arizona.edu...
> As an aside, do you avoid using BREAK instead of goto for compatibility
> with old IDL versions?
>

Actually, today I decided to set the minimum IDL version for my code to V5.4 instead of V5.3. V5.4 is when both BREAK and ARRAY_EQUAL() were introduced. So after first updating the the GOTOs to BREAK, I started to look if the code could be more efficient using ARRAY_EQUAL(). (I didn't immediately realize, though, that BREAK could be safely placed inside of an inner ENDIF clause, and it would still break out of the FOR loop.)

```
>> found = 1 - array_equal( array EQ value, 0)
```

```
> Just negate your ARRAY_EQUAL. I use this all the time:  
>  
>in_array=~array_equal(array NE value,1b)
```

This is still not quite optimal since the term (array NE value) is always evaluated for the entire array. That is why I was looking for a function, say, ARRAY_OR for which one could write

```
in_array = array_or(array,value)
```

which would return 1 as soon as it found an element of array equal to value, and not perform any operations on the remaining elements of the array.

--Wayne
