Subject: Re: Little help on arrays
Posted by JD Smith on Mon, 16 Feb 2004 19:23:45 GMT
View Forum Message <> Reply to Message

On Mon, 16 Feb 2004 09:27:52 -0600, Craig Markwardt wrote:

>
> Nuno Oliveira <nmoliveira@fc.ul.pt> writes:
>
>
>> How do I compare one array with another? I want to avoid comparing
>> position per position.
>>
>> IDL> if [1,1] eq [1,1] then print, 'bingo!'
>> % Expression must be a scalar or 1 element array in this context: <BYTE
>> Array[2]>.
>> % Execution halted at: $MAIN$
>
> The other posters have good ideas. My own idiom for this comparison
> is:
>
> if total(abs(X-Y)) EQ 0 then print, 'bingo!'

If you have a recent enough IDL, ARRAY_EQUAL() is the way to go,
because it stops the comparison as soon as it determines the arrays
are not equal, and the second array need not be an array, but can be a
scalar. I use this all the time for tricks like:

```
if ~array_equal(array ge 0,1b)
```

which efficiently determines if any element of array is not ge 0
(i.e. is negative). I also think it's a little more clear what the
intention is than TOTAL. Both the TOTAL and ARRAY_EQUAL method are
much faster than using WHERE, since they don't have to construct list
of indices. Craig's construct is very good when comparing floats, up
to some tolerance:

```
if total(abs(X-Y) ge 1.e-6) eq 0 then print, 'Close enough!'
```

but of course this can also be expressed as:

```
if array_equal(abs(X-Y) lt 1.e-6,1b) then print,'Close enough!'
```

which would run somewhat faster.

JD