
Subject: Re: subscript array question

Posted by [David Ritscher](#) on Thu, 11 Feb 1999 08:00:00 GMT

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

```
> array = intarr(5)
> subs = [0,2,4,4]
> array[subs] = array[subs] + 1
>
> and have the resulting values for array be:
>
>      1      0      1      0      2
>
> Because of the way IDL manages memory for expression evaluation
> and assignments, what happens for the last two elements of the
> addition is that the original value of array[4] is used twice,
> rather than what I want, which is to use the current value of
> array[4] each time. I.e. IDL gives the resulting values for
> array to be:
>
>      1      0      1      0      1
>
```

With my version, IDL Version 5.1.1, I get the latter, not the former! I suspect it is true with your version, as well. What IDL does is, for the duplicate subscript, it does the operation twice, but since 'array' on the right hand of the expression is a copy of the original, it goes and gets the same '0' twice, increments it by '1', and inserts it into the same location twice.

If in your actual application you're having similar problems, the `uniq` function might help you out:

```
array = intarr(5)
subs = [0,2,4,4]

subs = subs(uniq(subs, sort(subs)))

print, subs
      0      2      4

array[subs] = array[subs] + 1
```

In this case, it gives the same result, as I explained, but in your application, it might serve to solve your problem.

Good luck,

David Ritscher

--

Cardiac Rhythm Management Laboratory
Department of Medicine
University of Alabama at Birmingham
B168 Volker Hall - 1670 University Boulevard
Birmingham AL 35294-0019
Tel: (205) 975-2122 Fax: (205) 975-4720
Email: david.ritscher@bigfoot.com
