
Subject: Re: Evaluating a variable

Posted by [David Fanning](#) on Thu, 22 Jan 2004 18:04:27 GMT

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

Richard writes:

```
> I want to evaluate a (string) variable to then perform an operation on
> (another) variable whose name is the result of that evaluation.
>
> Let me clarify...
>
> I have a number of 1-D floating-point arrays. I want to take each array
> in
> turn and do stuff with it, something like this:
>
>     PRO blah
>
>     array1=[1,4,3,5.3,42,234,32,...(etc)]
>     array2=[1,6,4.65]
>     array3=[9.1,4,2.02,5,6]
>
>     arrays=['array1','array2','array3']
>     arrayno=-1
>
>     REPEAT BEGIN
>         arrayno=arrayno+1
>
> ; Pick which array I want to operate on, in order
> ; *****
>         thisarray=arrays(arrayno)
> ; *****
>
> ; Do stuff on the selected array
>         ROUTINE_DOSTUFF,thisarray
>
> ; Loop through all the arrays.
>         ENDREP UNTIL arrayno=2
>
>     END
>
> Except, of course, that this doesn't work, because (in the highlighted
> line) the variable 'thisarray' is set to be the string value 'array1'
> (etc), instead of having the actual array1 copied into it.
```

You could put your arrays into some kind of a list. (Something as simple as a pointer array might work, or you could use something like a linked list, if you want something more complicated.) I'd try something like this:

```
myArrays = PtrArr(3)
myArrays[0] = Ptr_New(array1, /No_Copy)
myArrays[1] = Ptr_New(array2, /No_Copy)
myArrays[2] = Ptr_New(array3, /No_Copy)
```

Then, when you want an array to work with:

```
thisarray = *(myArrays[arrayno])
```

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Evaluating a variable

Posted by [Richard](#) on Fri, 23 Jan 2004 09:35:02 GMT

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

David Fanning wrote:

```
> You could put your arrays into some kind of a list. (Something
> as simple as a pointer array might work, or you could use
> something like a linked list, if you want something more
> complicated.) I'd try something like this:
```

```
>
> myArrays = PtrArr(3)
> myArrays[0] = Ptr_New(array1, /No_Copy)
> myArrays[1] = Ptr_New(array2, /No_Copy)
> myArrays[2] = Ptr_New(array3, /No_Copy)
```

```
>
> Then, when you want an array to work with:
```

```
>
> thisarray = *(myArrays[arrayno])
>
```

Perfect - thank you very much.

Richard.
