Subject: Evaluating a variable
Posted by Richard on Thu, 22 Jan 2004 17:18:07 GMT
View Forum Message <> Reply to Message

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.

If I try this instead...

```
        arrays=[array1,array2,array3]

        thisarray=arrays(start,stop)
```

...this works, except I have to fiddle around calculating start and stop by finding the sizes of the array and all the preceding arrays, which isn't very nice.

Ahah, I think, I'll put them into one big 2-D array so I /know/ where each
one starts. But, alas, their lengths are very variable, so I'd end up with a
huge array, much of which would be 0, which seems a bit wasteful).

The same problem comes if I set up pointers to a structure:

```
 arraystruct={array1:PTR_NEW(array1),array2:PTR_NEW(array2).. .}

thisarray=arraystruct.arrayX ???
```

I'm having a brainfade here. What's the best way of doing this in IDL?
All I want to do is evaluate a string variable to read the value of the variable that it evaluates to (say that three times quickly)

Thanks in advance,
Richard
