

---

Subject: Re: Array elements as named variables?  
Posted by [thompson](#) on Mon, 27 Jun 1994 17:16:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

msegur@newton (The Ethereal Knight) writes:

> Is it possible to used an element of an array as a named variable  
> in order for a procedure to store a value in it?

The simple answer is no. Unfortunately, the inner mechanics of IDL doesn't allow you to do this. You have to do something like this

```
d_name = make_array(glob.ndims, /STRING)
d_size = make_array(glob.ndims, /LONG)
ncdf_diminq, cdfid, i, temp_name, temp_size
d_name(i) = temp_name
d_size(i) = temp_size
```

(assuming that the last two parameters of `ncdf_diminq` are output parameters.

Bill Thompson

---

---

Subject: Re: Array elements as named variables?  
Posted by [landers](#) on Mon, 27 Jun 1994 22:11:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <2un02p\$lm1@lace.Colorado.EDU>, msegur@newton (The Ethereal Knight) writes:

|> Is it possible to used an element of an array as a named variable  
|> in order for a procedure to store a value in it?  
[snip]

Nope, can't do it. Seems reasonable that you should be able to, since it points to a single memory location (at least at the time of the CALL, if not the RETURN, since the variable referenced could change during the subroutine processing!...), and you could have used it on the left-hand-side of an assignment. But...

When you call a procedure, IDL/WAVE decides whether to pass a parameter by value or by reference (memory address). There is no consideration given to making assignments upon a RETURN. Only things passed by reference can be changed, since the location of the actual variable in memory is the same in the caller and the subroutine.

The reasoning used to decide how to pass a parameter is if it's an expression or not. Check this by doing HELP (IDL) or INFO (WAVE) like this:

```
WAVE> info,a
A      FLOAT    = Array(10)
```

```
WAVE> info,a(0)
<Expression>  FLOAT   =   0.00000
^^^^^^^^^^
```

You want it to decide based on if you could use a reference on the left-hand side of an expression. Not me. The way it is, I can pass an array to a procedure like this:

```
Some_Procedure, a
```

or like this:

```
Some_Procedure, a(*)
```

The procedure gets the same data, but in the second case, I can insure that my data won't be changed behind my back. Just a matter of perspective.

;Dave

---

Subject: Re: Array elements as named variables?  
Posted by [jimbrakefd](#) on Tue, 28 Jun 1994 17:37:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <2un02p\$lm1@lace.Colorado.EDU>, msegur@newton (The  
Ethereal Knight) writes:

> Is it possible to used an element of an array as a named  
variable  
> in order for a procedure to store a value in it? When I try to do  
this, I  
> get an error telling me that the expression must be a named  
variable, but  
> it seems like a reasonable thing to want to do. I know that I can  
use a  
> dummy variable to store the value, but this seems awfully clumsy. Am  
I  
> missing something?

Of course one can pass the subscripted array as a text string for  
EXECUTE  
(.i.e. EXECUTE("A[2,5]" + "=" + "result")

Jim Brakefield

---

Subject: Re: Array elements as named variables?  
Posted by [schrick](#) on Tue, 28 Jun 1994 18:16:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The Ethereal Knight (msegur@newton) wrote:

: Is it possible to use an element of an array as a named variable  
: in order for a procedure to store a value in it?

No, but it is possible to pass whole arrays between procedures:

```
pro ncdf_diminq, cdfid, i, dname, dsize
  dname(i) = "testing 123"
  dsize(i) = 100
end

;main
  d_name = make_array(glob.ndims, /STRING)
  d_size = make_array(glob.ndims, /LONG)
  ncdf_diminq, cdfid, i, d_name, d_size
end
```

Mike

---