Vince Hradil (hradilv@yahoo.com) writes:

> I often have the need to replicate an array, but IDL's replicate only
> works with scalars. Does anyone have any tips on the most efficient,
> simplest, clearest (you choose) way to do this?

I am such a sucker for these kinds of articles. :-(

I don't know about "efficient, simplest, clearest", but
here is something that works for the three cases (1D, 2D,
and 3D arrays) I tried it on. I leave it to your imagination
to figure out the rest of the cases. It seemed like details
to me. :-)

Cheers,

David

```
 ************************************************************ ***********
FUNCTION Replicate_Array, array, number

  ; Obtain information about the array.

ndims = Size(array, /N_Dimensions)
dims = Size(array, /Dimensions)
type = Size(array, /TName)

  ; Initialize some variables.

snumber = StrTrim(number,2)
newArray = 0

  ; Create the new array and populate it with the old array.

CASE ndims OF
  1: BEGIN
    command = 'newArray = Make_Array(' + StrTrim(dims[0],2) + ',' $
      + snumber + ',' + type + '=1)'
    ok = Execute(command)
    FOR j=0,number-1 DO newArray[*,j] = array
    END
  2: BEGIN
    command = 'newArray = Make_Array(' + StrTrim(dims[0],2) + ',' + $
      StrTrim(dims[1],2) + ',' + snumber + ',' + type + '=1)'
```

```
      ok = Execute(command)
      FOR j=0,number-1 DO newArray[*,*,j] = array
      END
   3: BEGIN
      command = 'newArray = Make_Array(' + StrTrim(dims[0],2) + ',' + $
        StrTrim(dims[1],2) + ',' + StrTrim(dims[2],2) + ','  + snumber $
        + ',' + type + '=1)'
      ok = Execute(command)
      FOR j=0,number-1 DO newArray[*,*,*,j] = array
      END
   ELSE: Print, 'Sorry, figure this out yourself. :-)'
ENDCASE

RETURN, newArray
END
 *************************************************************** **********
```

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155