
Subject: A question to all you DLM experts...

Posted by [Randall Skelton](#) on Sat, 09 Jun 2001 16:31:54 GMT

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

Hi all,

There is something subtle that I'm missing here... I have a trivial DLM called 'array from nothing' because it is supposed to create an array in C using IDL temporary memory and then return it to IDL using VarCopy. The problem is, when I initially define my new array, it works fine, but that variable name seems to be defunct for re-assignment? What exactly does, 'Array has a corrupted descriptor' mean?

```
-----
IDL> ok=arrayfromnothing(a)
% Loaded DLM: MYIDL.
IDL> print, a
    0.00000    1.00000    4.00000    9.00000    16.0000
    25.0000    36.0000    49.0000    64.0000    81.0000
    100.000    121.000    144.000    169.000    196.000
    225.000    256.000    289.000    324.000    361.000
IDL> help, a
A          FLOAT    = Array[10, 2]
IDL> a=fltarr(10,2)
% Array has a corrupted descriptor: A.
% Execution halted at: $MAIN$
IDL> help, a
A          UNDEFINED = <Undefined>
-----
```

Below is the DLM code. Any comments?

Cheers,
Randall

```
-----
/* function: arrayFromNothing */
IDL_VPTR IDL_CDECL arrayFromNothing(int argc,
                                     IDL_VPTR argv[], char *argk) {
    /* Called in IDL as:
     *   ret = arrayFromNothing(new_array)
     */

    /* general index */
    int i;

    /* local pointer */
    float *ptr;
```

```

/* IDL specific */
IDL_MEMINT dim[2];
IDL_VPTR tmp;

/* set the correct dimension for the new array */
dim[0]=10;
dim[1]=2;

/* Make Sure we can write to it, free anything already associated */
IDL_StoreScalarZero(argv[0], IDL_TYP_FLOAT);

/* make a temporary IDL array with the same memory address as ptr */
ptr=(float *)IDL_MakeTempArray(IDL_TYP_FLOAT,2,dim,IDL_ARR_INI_ZERO,&am p,tmp);

/* fill the array */
for(i=0;i<100;i++) ptr[i]=i*i;

/* copy the IDL_VPTR tmp to passed arg */
IDL_VarCopy(tmp,argv[0]);

/* return 1 signifying no errors in execution */
return (IDL_GettmpLong(1));
}

```
