
Subject: Re: dlm creating an array?

Posted by [John-David T. Smith](#) on Tue, 03 Apr 2001 19:49:55 GMT

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

Randall Skelton wrote:

>
> Hello all,
>
> Is there a way to make a dlm that creates an IDL array?
>
> i.e. I am writing a simple file reading dlm that reads a binary file and
> generates a 1 dimensional array (or vector) and I would like this array
> returned to IDL such that the function syntax is something like:
>
> IDL> return = file_reader("test.bin", a);
>
> where "return" is for error checking, "test.bin" is the file to read, and
> "a" is what I would like the *new* array to be called.
>
> Is there something like, "IDL_StoreArray" as opposed to "IDL_StoreScalar?"
>
> Any and all hints and suggestions are greatly appreciated!

Two ways. Here is the better:

```
void test_array(int argc, IDL_VPTR argv[])
{
    float *test;
    int i;
    IDL_MEMINT dim[2];
    IDL_VPTR tmp;

    dim[0]=dim[1]=10;
    /* Make Sure we can write to it, free anything already associated */
    IDL_StoreScalarZero(argv[0], IDL_TYP_LONG);

    test=(float *)IDL_MakeTempArray(IDL_TYP_FLOAT,
    2,dim,IDL_ARR_INI_NOP,&tmp);
    IDL_VarCopy(tmp,argv[0]); /* This is the key. Copy tmp to passed arg
    */
    for(i=0;i<100;i++) test[i]=i*i;
}
```

That is, you make a temporary array, and copy it over to the passed argument (no data is actually copied, since it's a temporary). The StoreScalarZero makes sure it's passed by reference.

This method is easiest. Another way is to make your own data, and then use IDL_Import_Array to wrap an IDL_VPTR around it (also no copying performed). They are basically equivalent, but the typing is more up front with a the temporary variable method.

Please note that I did not free the tmp VPTR with IDL_DeTmp(). Why? Because it was already reclaimed by IDL_VarCopy (which in this case is more of a renaming than a copy). Doing so twice is a no-no.

What if you have an arbitrary number of dimensions (read from the file perhaps)? I'd simply declare dim[IDL_MAX_ARRAY_DIM] instead, and test to ensure this limit isn't surpassed.

Good luck,

JD

Subject: Re: dlm creating an array?
Posted by [Randall Skelton](#) on Wed, 04 Apr 2001 08:22:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the help JD and Jim! That was WAY easier than I thought it would be :)

Randall

```
> Two ways. Here is the better:
>
> void test_array(int argc, IDL_VPTR argv[])
> {
>   float *test;
>   int i;
>   IDL_MEMINT dim[2];
>   IDL_VPTR tmp;
>
>   dim[0]=dim[1]=10;
>   /* Make Sure we can write to it, free anything already associated */
>   IDL_StoreScalarZero(argv[0], IDL_TYP_LONG);
>
>   test=(float *)IDL_MakeTempArray(IDL_TYP_FLOAT,
> 2,dim,IDL_ARR_INI_NOP,&tmp);
>   IDL_VarCopy(tmp,argv[0]); /* This is the key. Copy tmp to passed arg
> */
>   for(i=0;i<100;i++) test[i]=i*i;
> }
>
> That is, you make a temporary array, and copy it over to the passed
```

> argument (no data is actually copied, since it's a temporary). The
> StoreScalarZero makes sure it's passed by reference.
>
> This method is easiest. Another way is to make your own data, and then
> use IDL_Import_Array to wrap an IDL_VPTR around it (also no copying
> performed). They are basically equivalent, but the typing is more up
> front with a the temporary variable method.
>
> Please note that I did not free the tmp VPTR with IDL_DeITmp(). Why?
> Because it was already reclaimed by IDL_VarCopy (which in this case is
> more of a renaming than a copy). Doing so twice is a no-no.
>
> What if you have an arbitrary number of dimensions (read from the file
> perhaps)? I'd simply declare dim[IDL_MAX_ARRAY_DIM] instead, and test
> to ensure this limit isn't surpassed.
>
> Good luck,
>
> JD
>
