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_DelTmp().  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
>