
Subject: IDL-Fortran String/Char Passing
Posted by [wicker](#) on Tue, 15 Jun 1993 19:01:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi there. We are doing some idl development which is using some fortran routines. Does anyone out there know how to pass character data from idl to fortran and vice versa? I know how to do integer and real data, no problem. Examples of working code would be VERY welcome.

Also, anybody else notice that the new HDF interface in IDL is REALLY slow as compared to C or Fortran.

Thanks in advance.

```
|
| Prof. Louis Wicker      email: wicker@ariel.tamu.edu |
| TAMU/Meteorology Dept  phone: (409) 845 - 8084   |
| 1204 Eller O&M Bld     fax: (409) 826 - 4466    |
| College Station TX 77843-3150                    |
```

Subject: Re: IDL-Fortran String/Char Passing
Posted by [Bill Davis](#) on Wed, 16 Jun 1993 16:41:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <wicker-150693135755@wicker.tamu.edu> Dr. Louis J. Wicker, wicker@ariel.tamu.edu writes:

> Hi there. We are doing some idl development which is using some fortran
> routines. Does anyone out there know how to pass character data from idl
> to fortran and vice versa? I know how to do integer and real data, no
> problem. Examples of working code would be VERY welcome.

I do it under VMS. Read all about CALL_EXTERNAL in the User's Guide and Reference Manual. Passing characters into a FORTRAN routine is no problem. I must say that it was not trivial to implement returning strings even after reading the manual. I feel something like the following example should be included in the documentation:

VMS FORTRAN routine to return a string variable via IDL CALL_EXTERNAL:

```
      INTEGER* 4 FUNCTION retstring      ! address pointer is  
returned
```

```
      CHARACTER*16 local_string  
      CHARACTER*1 null  
      DATA null /'00'X/
```

```
    local_string = 'new stuff'//null    ! Concatenate null
terminator
    retstring = %LOC(local_string)      ! Return address of string

RETURN
END
```

This routine is put into a VMS shared image which is pointed to by the logical name MY_IMAGE (also non-trivial if you have never done it). Then in IDL:

```
strvar=CALL_EXTERNAL(/S_VALUE,'MY_IMAGE','RETSTRING')
```

This is painfull if you have more than 1 string variable (I believe you will need one routine per string variable). You may wish to see if using LINKIMAGE would be easier.
