

---

Subject: Tutorial on call\_external  
Posted by [gotwols](#) on Tue, 29 Jan 1991 20:48:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I have been asked by several people to post an example that illustrates how to "hook" a program into IDL from another language. The example below was copied from the IDL Users Manual (with some poetic license). It illustrates how to call a Fortran program to average an array of data. This is a fairly typical example which illustrates how IDL passes numeric values by reference. This is the natural passing mode in FORTRAN so you don't have to do anything special. In C however you have to remember to use pointers. I'll post a simple C example in the near future. There is one other important thing that is not illustrated below. Passing strings must be done differently. Again, I'll post an example of how to pass strings soon.

This example has only been tested under VMS. As of the latest version of IDL (2.0.13) SUN users also have call\_external capability. It would be nice if a SUN user would post a tutorial file similar to this one that shows ALL of the steps needed to run this program on a SUN.

You should be able to cut and paste the three brief listings below into files my\_avg.for, build\_my\_avg.com, and call\_extern\_ex.pro, respectively. Read carefully the notes in call\_extern\_ex.pro about the logical symbol my\_avg\_exe.

-- Bruce

\*\*\*\*\* Here is the FORTRAN Program: my\_avg.for \*\*\*\*\*

```
REAL * 4 function MY_AVG(V,N)
REAL*4 V(*)
MY_AVG = 0.0
DO I=1,N
  MY_AVG = MY_AVG + V(I)
END DO
MY_AVG = MY_AVG / N
RETURN
END
```

\*\*\*\*\* Here is the command file to compile the FORTRAN program\*\*\*\*\*

\*\*\*\*\* and define a logical name that points to the executable\*\*\*\*\*

\*\*\*\*\* I call this build\_my\_avg.com and invoke it by typing\*\*\*\*\*

\*\*\*\*\* @build\_my\_avg \*\*\*\*\*

```
$ FORTRAN MY_AVG
$ LINK MY_AVG, SYS$INPUT/OPT/SHARE
UNIVERSAL = MY_AVG
$! Replace D0:[GOTWOLS.IDL.FOR] below with the directory where your EXE file
```

```
$! resides
$ DEFINE MY_AVG_EXE D0:[GOTWOLS.IDL.FOR]MY_AVG.EXE
```

```
*****Here is an IDL "wrapper program that takes care of all of *****
***** the nonsense you need to remember about call_external *****
***** I always use such wrapper routines as a matter of style*****
; call_extern_ex.pro -- Example using call_external based on the IDL
; Users Guide , pg. 16-15, Edition of June 8, 1990. B. L. Gotwols jhu/apl
;
; To run this example all you need to do is type call_extern_ex and if
; everything works right it will print out "mean = 4.5000".
```

```
pro call_extern_ex,meanx
```

```
; Use call_external in IDL v2.0.13 and later. In earlier versions you
; must use call_vms instead. (But call_vms is only available under VMS.)
; Note: 1. /f_value tells call_external to return a float rather than a
;        32 bit integer.
;        2. my_avg_exe is a logical symbol that points to my_avg.exe.
;        Alternatively you can replace 'my_avg_exe' with 'my_avg' and
;        then place my_avg.exe in sys$common:[syslib]. (The IDL
;        manual says to use sys$share but this is a different place for
;        each member of our cluster, hence I prefer to use sys$common)
meanx = call_external(/f_value, 'my_avg_exe', 'my_avg', $
    findgen(10), 10L) ; Note: The IDL manual forgot the $
```

```
print,'mean = ',meanx
return
end
```

```
--
```

```
Bruce L. Gotwols
Johns Hopkins University, Applied Physics Lab., Laurel MD 20723
Internet: gotwols@warper.jhuapl.edu (128.244.176.48)
SPAN: APLSP::STR::GOTWOLS
```

---