
Subject: Re: CALL_EXTERNAL simple problem ?
Posted by [Randall Skelton](#) on Wed, 14 Nov 2001 13:17:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

A few important suggestions:

- 1) Put away all sharp objects (scissors, pencils, etc.)
- 2) Read through the entire external development guide.
- 3) Read through the entire external development guide (again!).
- 4) Send RSI a feature request for providing asking for C API documentation for heap variable access (ok, so this is one is for my own personal benefit).
- 5) Buy Ronn's book on calling C from IDL using DLMs.

If you are convinced that call_external is what you want, try:

```
-- test.c --  
void test(int argc, void* argv[])  
{  
    float* f;  
    f = argv[0];  
    f[0] = 0.0;  
}  
--
```

compile with something like: 'gcc -fpic -shared -o test.so test.c'

```
-- test.pro --  
PRO test  
f=[1.0,1.0]  
PRINT,f  
s=call_external('test.so','test',f)  
PRINT,f  
END  
--
```

On 14 Nov 2001, trouble wrote:

```
> Hi,  
>  
> OK - I've finally bitten the bullet and am calling C from IDL. So far,  
> it hurts a lot. I have the following simple code which gives a seg.  
> fault and I'm really hoping someone can help !  
>  
> C code  
> -----  
> void test(float f[])
```

```
> {
> f[0]=0.0;
> }
>
>
> IDL code
> -----
> PRO test
> f=[1.0,1.0]
> s=call_external('image.so','test',f)
> PRINT,f
> END
>
>
>
> The C code compiles alright using the same flags as in the
> CALL_EXTERNAL examples, which incidentaly work fine. I'm using IDL5.4
> under Linux on Alpha.
>
> Thanks,
> Ciao.
>
```

Subject: Re: CALL_EXTERNAL simple problem ?
Posted by [the_cacc](#) on Thu, 15 Nov 2001 12:58:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
>
> If you are convinced that call_external is what you want, try:
>
```

I really didn't want to do it - some nasty boys here called me bad names for criticizing IDL because it didn't have the functions I want, so I had no choice...

```
>
> void test(int argc, void* argv[])
>
```

Yes - this works! It seems that IDL (on my platform only?) needs C functions to have the (argc,*argv[]) form, despite examples in the manual to the contrary. However, now I understand.

Ciao.

Subject: Re: CALL_EXTERNAL simple problem ?
Posted by [Randall Skelton](#) on Thu, 15 Nov 2001 14:51:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 15 Nov 2001, trouble wrote:

[snip]

> Yes - this works! It seems that IDL (on my platform only?) needs C
> functions to have the (argc,*argv[]) form, despite examples in the
> manual to the contrary. However, now I understand.

Glad it works... but which manuals are you using? All platforms I can think of would require such a construct to pass IDL variables into C. Moreover, all of the examples I can find in chapter 7 of the IDL 5.3 External Development Guide (EDG) show the C routines to have the form:

```
function_name (int argc, void* argv[])
```

where:

'argc' is the argument counter (i.e. the number of arguments including the name of the call itself)

'argv' is the argument vector which is an array of pointers to the passed IDL variables, which are accessible via argv[0] ... argv[argc-1]. Note that argv has one additional element, argv[argc] which is always a null pointer. You should be very careful to appropriately define your C variables so they match the size/structure of the internal IDL variables (defined in export.h). If you have a mismatch in the size of your variables, you will corrupt your IDL session.

The real 'root' of the (argc, argv) construction likely arises from the way that a stand-alone C/Fortran program accept simple command-line arguments.

Cheers,
Randall
