Subject: CALL_EXTERNAL
Posted by dean on Wed, 18 Aug 1993 16:38:08 GMT
View Forum Message <> Reply to Message

I am using CALL_EXTERNAL to call a C routine to open a certain file with
a special image format. However, the C routine is not getting the right
file name which is a character string. A FORTRAN routine gets the right
character string from CALL_EXTERNAL, but the C routine doesn't. Can someone
provide me with some hints on how to get IDL to pass the right character
string to my C routine through CALL_EXTERNAL?

*** \ *****
Kelly Dean \
Cooperative Institute for Research in the Atmosphere ___\
Foothills Campus /|
Colorado State University / |\
Fort Collins, CO 80523 / | \
E-MAIL: DEAN%SOL.DNET@SIRIUS.CIRA.COLOSTATE.EDU | \
** **********

Subject: Re: Call_external
Posted by Inigo Garcia on Thu, 12 Jun 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Harald Frey wrote:
>
> I have changed from VMS to UNIX and now I got a problem with
> call_external and
> I'm not sure if it is an IDL or a UNIX problem. I work under
> { sparc sunos unix 4.0.1c}.
>
> The FORTRAN program example.f according to the User's Guide 18-14
>
> C -------------------------
> subroutine sum_array(array,n,sum)
>
> integer*4 n
> real*4 array(n),sum
>
> sum=0.0
> do i=1,n
> sum=sum+ array(i)
> print*, sum, array(i)
> enddo
>
> return
> end

> C -------------------------
>
> And the IDL program test_call.pro
>
> ; ------------------------
> pro test_call
>
> x=findgen(10)
> s=call_external('example.so','_sum_array_',x,n_elements(x),/ f_value)
>
> end
> ; ------------------------
>
> Compiling the FORTRAN program with (User's Guide 18-21)
>
> f77 -c -pic example.f
> ld -o example.so example.o
>
> Gives the error message
>
> % ld -o example.so example.o
> Undefined   first referenced
> symbol   in file
> __e_wsle   example.o
> __s_wsle_nv   example.o
> __do_l_out   example.o
> ld: fatal: Symbol referencing errors. No output written to example.so
>
> Trying
> ld -G -o example.so example.o
>
> compiles without error message but then in IDL I get
>
> IDL>.r test_call
> IDL> test_call
> % CALL_EXTERNAL: Error loading sharable executable.
>   Symbol: _sum_array_, File = example.so
>   ld.so.1: /usr/local/idl/bin/bin.solaris2/idl: fatal:
> dlsym:
>   can't find symbol: _sum_array_
> % Execution halted at: TEST_CALL   4 test_call.pro
> % $MAIN$
>
> Where is the problem?
>
> Harald Frey
> Space Sciences Laboratory
> University of California

> Berkeley, CA 94720
> hfrey@ssl.berkeley.edu

Hi Harald:

The compiling to use a Fortran subroutine from IDL is the second one you mention, the "ld -G -o example.so example.o". The error you get is due to the fact that IDL does not find the entry "_sum_array_" in the file "example.so". The most probable thing is that the entry is called "sum_array_", at least in Solaris is that way. To verify the enties that one of your subroutines have, you can use the command "nm example.o".

Anyway, with your program written like this nothing is going to work, because you need to use an Interface between IDL and the Fortran Routine. How to do this is explained in the Online Help of IDL, in the Advanced Development Guide, CALL_EXTERNAL Section. The example files are in the /external/sharelib/ directory of the main IDL directory.

Enjoy it !!

I~nigo.

--
\\|//
(o o)
+----------oOOo-(_)-oOOo---------- --+
| I~nigo Garcia Ruiz |
| Kapteyn Instituut Phone: +31-(0)50-3634083 |
| Landleven 12 Fax: +31-(0)50-363 |
| 9747 AD GRONINGEN (Netherlands) e-mail: iruiz@astro.rug.nl |
+---------- --+

Subject: Re: call_external
Posted by Vap User on Wed, 05 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Chris Varekamp <chris.varekamp@users.whh.wau.nl> writes:

The entry point is the name of the C function in the source file.

The function ...

```
float foo ( int argc, char *argv[] ) {
/* your code here */
return (result);
}
```

within the file 'foo.c' would be called in the following manner using
call_external.

  result = call_external('c:\path\to\shareable\library\foo.ddl','foo',arg1,...)

I may have made a mistake about the naming convention of the 'image'
(i.e. the library/ddl file) since I work with Unices and don't really
know the nomenclature for NT boxes, but I think a DDL is the
equivalent of a Unix shareable library on a windoze machine.

In any case, the 'entry' is just the name of the C function. It's
called an entry point because you can have more than one function in
the same shareable image, hence the loader must know at which point to
'enter' the file.

>
> Hello,
>
> Does anyone have experience running a C program using the CALL_EXTERNAL
> function in IDL on an NT machine? Are there examples around ?
>
> I have a small program from numerical recipes that returns a float.
> However, the CALL_EXTERNAL function requires you to specify an entry
> point (don't know what is meant by this).
>
> Hope someone can give me a hint...
>
> Chris Varekamp
> Dept. Water Resources
> Wageningen University
> The Netherlands

--
I don't speak for JPL, it doesn't speak for me.
Well, not all the time, at least.
William Daffer <vapuser@haifung.jpl.nasa.gov>

Subject: Re: call_external
Posted by rivers on Wed, 12 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <35C84E51.47460D9@users.whh.wau.nl>, Chris Varekamp
<chris.varekamp@users.whh.wau.nl> writes:
> Hello,
>
> Does anyone have experience running a C program using the CALL_EXTERNAL

> function in IDL on an NT machine? Are there examples around ?

The IDL 5.1 distribution directory includes an example which works under NT in IDL51/external/sharelib

NOTE: This example does not work correctly in passing strings. RSI changed the way strings are passed with CALL_EXTERNAL under IDL 5.1, but they did not change the example program.

Mark Rivers (773) 702-2279 (office)
CARS (773) 702-9951 (secretary)
Univ. of Chicago (773) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory (630) 252-0422 (office)
Building 434A (630) 252-0405 (lab)
9700 South Cass Avenue (630) 252-1713 (beamline)
Argonne, IL 60439 (630) 252-0443 (FAX)

Subject: Re: CALL_EXTERNAL
Posted by menakkis on Mon, 22 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

> Hello!
> Help me please!
> I called function CALL_EXTERNAL and recieved this message:
> ERROR_MOD_NOT_FOUND
> What is it mean and what's my mistake?
> code is:
> CALL_EXTERNAL('DllProject.dll','bit_operation',buffer[pos+6])
> output:
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: bit_operation, File = ./DllProject.dll
> ERROR_MOD_NOT_FOUND
> in dll exist export function with name "_bit_operation"(checked in QuikView)

Sounds like you'd be quite pleased to get this working :-) Sorry for the delay, but there's been a week-end. (In other words, my apologies to the NG for my previous flippant post.)

I assume that you're using C. In case you haven't got it all working yet, the short answer is: Declare your function as WINAPI (which translates to __stdcall), and export it with a .DEF file, like in the example that comes

with IDL. (You have to do a little digging to discover that __stdcall is used in the examples.) If you're using visual C then just plonk the .DEF file amongst your c and h files via "Add to project" -> "Files...".

Now there are other ways to get CALL_EXTERNAL to work, especially since IDL started supporting the "cdecl" calling convention as an option. Myself, I haven't experimented in this area for a while, as I found the .DEF file method works in all IDL versions that are win32-based. But here are some suggestions... If you don't mind making your IDL code more Windows-specific, you could alternatively try sticking an underscore on the front of the function's name in your IDL code. If the function has been exported with only this name decoration AND (somehow) built to use the "stdcall" calling convention then it should work. If the "cdecl" calling convention, then you'll also have to use the /CDECL switch in CALL_EXTERNAL(). If you have exported your function by declaring it as __declspec(dllexport) (which has been the Microsoft- recommended way for quite some time now) then I think it'll use cdecl, but I'm not really sure. (Some dark corners of the MS documentation allude to the calling convention being optional here, like it'll go with the default if you don't add "__stdcall". It's rather obscure :-( ) I also don't know whether IDL will do you the favour of trying a leading underscore with /CDECL if the plain name doesn't work - it should do as a leading underscore is cdecl's default way of decorating names, apparently. Give it a try.

Peter Mason

Subject: Re: CALL_EXTERNAL
Posted by ashmall on Wed, 03 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

philf@astro.lsa.umich.edu (Phil Fischer) wrote:

> After issuing a call_external command, it seems to me that the called program
> (*.so file) remains in memory until idl is exited. Is there any way of freeing
> it from memory?

Unfortunately I think this is just the way it is, which is obviously a pain for debugging the externally-called program. Having said this my experience is of dll's under NT.

Justin

Subject: Re: CALL_EXTERNAL
Posted by Craig Markwardt on Fri, 19 Jul 2002 22:00:34 GMT
View Forum Message <> Reply to Message

"Ian Dean" <Ian.d.dean@baesystems.com> writes:

> Hi,
> I'm running IDL 5.4 under VMS (!!!), and would like some help using
> CALL_EXTERNAL to a C routine.
>
> The C routine has a sinle parameter passed to it, but it is a structure
> (similar to that below)
> The routine returns to the caller a variable sized array starting at the
> address of item buffer and the number of elements in buff_size. (Other
> control fields are also used but I won't cloudy the water with these).

The answer to your question is simple. You can't use CALL_EXTERNAL to pass IDL structures to an external routine. In fact, I don't believe you can do this with the DLM or LINK_IMAGE facilities either. IDL structures are intentionally a black box not meant to be interrogated.

Also, it's not as straightforward as you think to pass raw memory, such as your "void *buffer", to and from IDL. You'll have to investigate the External Developers Guide and find out how to insert user data into an IDL variable.

The much easier approach is to make two wrapper routines, written in both IDL and C. On the IDL side, this routine would transfer IDL structure values into simple IDL variables which can then be passed easily to CALL_EXTERNAL. On the C side, you would need a similar wrapper which would pack those values back into your own C structure, and proceed on your merry way.

Good luck,
Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: CALL_EXTERNAL
Posted by ronn on Sat, 20 Jul 2002 13:25:32 GMT
View Forum Message <> Reply to Message

in article onofd3ml8d.fsf@cow.physics.wisc.edu, Craig Markwardt at

craigmnet@cow.physics.wisc.edu wrote on 7/19/02 6:00 PM:

>
>
> "Ian Dean" <Ian.d.dean@baesystems.com> writes:
>
>> Hi,
>> I'm running IDL 5.4 under VMS (!!!), and would like some help using
>> CALL_EXTERNAL to a C routine.
>>
>> The C routine has a sinle parameter passed to it, but it is a structure
>> (similar to that below)
>> The routine returns to the caller a variable sized array starting at the
>> address of item buffer and the number of elements in buff_size. (Other
>> control fields are also used but I won't cloudy the water with these).
>
> The answer to your question is simple. You can't use CALL_EXTERNAL to
> pass IDL structures to an external routine. In fact, I don't believe
> you can do this with the DLM or LINK_IMAGE facilities either. IDL
> structures are intentionally a black box not meant to be interrogated.
>
Hi Ian,

Craig is correct in that you can't use CALL_EXTERNAL to pass a structure, but you CAN use a DLM to do it. The details are in my "IDL calling C" book, but the trick is to create a structure in C and mimick it exactly on the IDL side. This means IDL ints must be C shorts, etc.

I also agree that you don't want to pass raw memory from C to IDL. IDL just doesn't know what to do with a memory address. Instead you would want to create an IDL structure on the C side and fill one of the fields with your data. Then you can use the structure in IDL just like normal.

Hope that helps a little.

-Ronn

--
Ronn Kling
KRS, inc.
email: ronn@rlkling.com
"Application Development with IDL"ï¿½ programming book updated for IDL5.5!
"Calling C from IDL, Using DLM's to extend your IDL code"!
http://www.rlkling.com/

Subject: Re: CALL_EXTERNAL
Posted by Nigel Wade on Mon, 22 Jul 2002 08:44:30 GMT
View Forum Message <> Reply to Message

Ian Dean wrote:

> Hi,
> I'm running IDL 5.4 under VMS (!!!), and would like some help using
> CALL_EXTERNAL to a C routine.
>
> The C routine has a sinle parameter passed to it, but it is a structure
> (similar to that below)
> The routine returns to the caller a variable sized array starting at the
> address of item buffer and the number of elements in buff_size. (Other
> control fields are also used but I won't cloudy the water with these).
>
> typedef struct
> {
> void *buffer;
> int buff_size;
> }ACCESS
>
> This routine and structure are already in use between several other C
> routines. I just want to use the same idea in IDL.

Unfortunately that won't be possible as IDL has no concept of void pointers. Even the pointers it does have are pointers to IDL variables, not raw memory addresses (at least I think that's correct).

What is that you are trying to achieve?

If you want to access the data in IDL then you'll have to create a variable of the appropriate type and either use your data from C as its data, or copy it, whichever you prefer.

If you want to pass on this structure to some other C function then you will need to create an IDL variable which can hold the pointer.

The former requires a DLM (or LINK_IMAGE) whilst the latter could be achieved via CALL_EXTERNAL provided you can be certain how much memory is required to hold your void pointer.

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: CALL_EXTERNAL
Posted by Mark Rivers on Wed, 31 Jul 2002 02:26:02 GMT
View Forum Message <> Reply to Message

Zakaria Aygula <zaygula@med.miami.edu> wrote in message
news:c934bc5c.0207300643.123066ce@posting.google.com...
> Hi all,
>
> I am not able to pass below strings from IDL 5.5 to c/c++ interface
> running on Red Hat Linux 7.3 2.96-110. The same code worked well on
> IDL 5.4 on solaris
> The fragment of the code is shown below.
>
> FUNCTION vgam_wrap_do_query, query, dest
> FORWARD_FUNCTION lib_name, lib_func_name

The way IDL passes strings to CALL_EXTERNAL has changed between IDL 5.4 and
IDL 5.5, because the definition of an IDL_STRING structure in export.h has
changed. This is the IDL 5.5 definition:

```
typedef int IDL_STRING_SLEN_T;
#define IDL_STRING_MAX_SLEN 2147483647
typedef struct {              /* Define string descriptor */
  IDL_STRING_SLEN_T slen;     /* Length of string, 0 for null */
  short stype;                /* type of string, static or dynamic */
  char *s;                    /* Addr of string */
} IDL_STRING;
```

In IDL 5.4 IDL_STRING_SLEN_T was "short", now it is "int", so strings can be
much longer now. CALL_EXTERNAL will still work, but you need to recompile
your shareable library with the new version of export.h

Mark Rivers

Subject: Re: CALL_EXTERNAL
Posted by zaygula on Wed, 31 Jul 2002 14:35:51 GMT
View Forum Message <> Reply to Message

zaygula@med.miami.edu (Zakaria Aygula) wrote in message
news:<c934bc5c.0207300643.123066ce@posting.google.com>...
> Hi all,
>
> I am not able to pass below strings from IDL 5.5 to c/c++ interface
> running on Red Hat Linux 7.3 2.96-110. The same code worked well on
> IDL 5.4 on solaris
> The fragment of the code is shown below.
>
> FUNCTION vgam_wrap_do_query, query, dest

```
>  FORWARD_FUNCTION lib_name, lib_func_name
>
>  IPaddress = 'localhost'
>  DBname    = 'gas'
>  DBlogin   = 'ziko'
>  DBpasswd  = 'tabes'
>  rc = call_external(lib_name('gavaidl'), $
>                     lib_func_name('gava_idl_do_query'),$
>         query, dest, n_elements(dest), $
>         IPaddress,  $
>         DBname,    $
>         DBlogin,   $
>         DBpasswd,  $
>         /i_value,  $
>         /portable )
>  return, rc
>
>  end
>
>  IDL>dest=strarr(16)
>  IDL>query='SELECT * FROM table1'
>  %Compiled module: LIB_NAME.
>  %Compiled module: LIB_FUNC_NAME.
>
>  printed results from c/c++ interface
>
>  cout<<IPaddress[0].s<<endl; = (null) expected output 'localhost'
>  cout<<DBname[0].s   <<endl; = (null) expected output 'gas'
>  cout<<DBlogin[0].s  <<endl; = (null) expected output 'ziko'
>  cout<<DBpasswd[0].s <<endl; = (null) expected output 'tabes'
```

After building up the shared object files with "export.h" of IDL_5.5,
then "cout<<IPaddress[0].slen<<endl; etc.. " gave correct string length
but "cout<<IPaddress[0].s<<endl; etc.." still gives "null" output.

Subject: Re: CALL_EXTERNAL
Posted by Randall Skelton on Wed, 05 Feb 2003 08:47:33 GMT
View Forum Message <> Reply to Message

Hi Thomas,

IDL pointers != C pointers

In short, you cannot pass an IDL pointer into C and expect to do anything
useful with it. You most certainly cannot pass an IDL pointer to C,
operate on it, and then pass it back with useful data. If you would like
this functionality added to IDL, please write support@rsinc.com and add

your name to the growing list of people wanting the C API to the heap variable and objects. In reality, however, this is probably not the functionality that you need.

What you are trying to do is a little outside the scope of what call external is truely useful for. You should probably consider the purchase of Ronn Kling's book on calling C from IDL using DLMs:

http://www.kilvarock.com/books/callingCfromIDL.htm

I'm not sure how feasible it is using CALL_EXTERNAL, but the only way you might be able to do this would involve passing an undefined variable (idl type=0) and allocate the memory directly using C. Check out the middle of Chapter 9 (External Development Guide) for the C functions that can help in this regard (IDL_MakeTempArray, IDL_ImportArray, IDL_ImportNamedArray). Once you have allocated the IDL array variable you then need to tie it to the passed (undefined) variable using IDL_VarCopy.

Again, the best way to do this is via a DLM and Ronn's book is, by far, the best way to learn.

Cheers,
Randall

On Wed, 5 Feb 2003, Thomas Gutzler wrote:

> Hi again,
>
> please correct me if I'm wrong.
>
> CALL_EXTERNAL
> - needs the /CDECL keyword to call a dll built by Borland C++ Compiler
> with: extern "C" __declspec(dllexport)
> It works with and without /CDECL for me
> - can have values and references (default) as arguments but cannot
> RETURN a pointer. Just a scalar variable with the value of an address
> - cannot have a NullPointer as argument and receive a Pointer to a
> variable/array (see below)
>
> What I'm trying to do is:
>
> C function:
> int test(int argc, void* argv[])
> {
> if(argc != 2) return 0;
> IDL_UINT *size = (IDL_UINT *) argv[0];
> int *array = (int *) argv[1];

```
>          array = new int[*size];
>          for (int i = 0; i < *size; i++) array[i] = i;
>          return 1;
> }
>
> IDL:
> IDL> array = PTR_NEW()
> IDL> Result = CALL_EXTERNAL('mydll.dll', '_test', 10, array, /CDECL)
> IDL> print, array
> should be
>       0     1     2     3     4     5     6     7     8     9
> but array is still a NullPointer
>
> If I leave the line
>          array = new int[*size];
> I have to initialize with array=intarr(10) and the returned array is:
>       0     0     1     0     2     0     3     0     4     0
>
> This is not what I expected (funny values! maybe type-conversion helps?)
> and not what I want. I want to receive a pointer to an array, because
> IDL doesn't know the size of the array being returned (Yes, I could
> allocate a 1000000x1 array and  resize it).
>
> Is this possible ?
>
> Tom
>
>
```

Subject: Re: Call_external
Posted by Dr Paul Ducksbury on Wed, 10 Mar 2004 13:02:56 GMT
View Forum Message <> Reply to Message

In addition to this query, ive now found an old version (5.0) of the
development guide but it just says for building a shareable object for
windows go get a windows programming guide, which isnt much help.
Does anyone know where i can find simple info on how to build a shareable
object withinh visual c++ 6.0 ?

thanks

"Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
news:1078919071.346356@bengal...
> Is there a simple guide to using call_external ? as i dont have the
external
> develpoment manual that the help page suggests.
>

> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>
> thanks
>
> Paul
>
> --
> ---------- ----------
--
> --
> This e-mail may contain information which is commercial-in-confidence
and/or
> legally privileged. For persons other than the intended recipient(s), any
> use, disclosure, copying or distribution of the e-mail or information
> contained therein is prohibited
> ---------- ----------
--
> --
> Dr P.G.Ducksbury BSc,PhD,CEng,FBCS Phone: (+44) 1684-895742
> E311, Fax:
> (+44) 1684-894952
> QinetiQ, Malvern Technology Center,
> St Andrews Road, Great Malvern, Worcestershire, WR14 3PS, UK.
> Email: p.ducksbury@signal.qinetiq.com
> ---------- ----------
--
> --
>
>
>

Subject: Re: Call_external
Posted by Haje Korth on Wed, 10 Mar 2004 13:11:37 GMT
View Forum Message <> Reply to Message

Paul,
Checkout the examples in the RSI\IDL60\external directory. They are pretty self-explanatory and are probably as descriptive as it will ever be. The only book on external routines that I know of, was published by Ronn Kling and specializes on DLMs. While DLM are extremely nice to handle within IDL. They are not the easiest to implement.

Cheers,
Haje

"Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
news:1078919071.346356@bengal...
> Is there a simple guide to using call_external ? as i dont have the
external
> develpoment manual that the help page suggests.
>
> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>
> thanks
>
> Paul
>
> --
> ---------- ----------
--
> --
> This e-mail may contain information which is commercial-in-confidence
and/or
> legally privileged. For persons other than the intended recipient(s), any
> use, disclosure, copying or distribution of the e-mail or information
> contained therein is prohibited
> ---------- ----------
--
> --
> Dr P.G.Ducksbury BSc,PhD,CEng,FBCS Phone: (+44) 1684-895742
> E311, Fax:
> (+44) 1684-894952
> QinetiQ, Malvern Technology Center,
> St Andrews Road, Great Malvern, Worcestershire, WR14 3PS, UK.
> Email: p.ducksbury@signal.qinetiq.com
> ---------- ----------
--
> --
>
>
>

Subject: Re: Call_external
Posted by Pepijn Kenter on Wed, 10 Mar 2004 13:19:12 GMT
View Forum Message <> Reply to Message

Dr Paul Ducksbury wrote:
> Is there a simple guide to using call_external ? as i dont have the external
> develpoment manual that the help page suggests.
>
I think it's only delivered in electronic form in IDL 6. Look for a file
called edg.pdf

HTH, Pepijn.

Subject: Re: Call_external
Posted by Dr Paul Ducksbury on Wed, 10 Mar 2004 14:39:28 GMT
View Forum Message <> Reply to Message

Thanks for that it looks easier than MS description of building shareable objects and def files. Only one small problem i'm having now is that all_callext_examples fails as it was unable to find some files. I assume its some problem so obvious its hitting me in the face, just cant see it.

Anyone any ideas ?

thanks

```
IDL> all_callext_examples
% Compiled module: ALL_CALLEXT_EXAMPLES.
% Compiled module: SIMPLE_VARS.
Calling simple_vars with the following arguments:
B_L             BYTE      =    2
I_L             INT       =       3
L_L             LONG      =           4
F_L             FLOAT     =      5.00000
D_L             DOUBLE    =       6.0000000
% Compiled module: GET_CALLEXT_EXLIB.
The name specified is not recognized as an
internal or external command, operable program or batch file.
cl  -D_DLL -DWIN32 -D_MT /nologo /I"E:\RSI\IDL60\external\include" /c
"E:\RSI\IDL60\external\call_external\C\incr_struct.c"
/Fo"incr_struct_166_PDPC.obj"
Could Not Find
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
external_examples_166_PDPC.exp
Could Not Find
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
external_examples_166_PDPC.lib
Could Not Find
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\incr_
struct_166_PDPC.obj
Could Not Find
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\simpl
e_vars_166_PDPC.obj
Could Not Find
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\strin
g_array_166_PDPC.obj
Could Not Find
```

C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_a
rray_166_PDPC.obj
Could Not Find
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_2
d_array_166_PDPC.obj
% CALL_EXTERNAL: Error loading sharable executable.
Symbol: simple_vars, File =
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
external_examples.dll
ERROR_MOD_NOT_FOUND
% Error occurred at: SIMPLE_VARS 91
E:\RSI\IDL60\external\call_external\C\simple_vars.pro
% ALL_CALLEXT_EXAMPLES 58
E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
% $MAIN$
% Execution halted at: ALL_CALLEXT_EXAMPLES 58
E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro

"Haje Korth" <haje.korth@jhuapl.edu> wrote in message
news:c2n469$63v$1@aplcore.jhuapl.edu...
> Paul,
> Checkout the examples in the RSI\IDL60\external directory. They are pretty
> self-explanatory and are probably as descriptive as it will ever be. The
> only book on external routines that I know of, was published by Ronn Kling
> and specializes on DLMs. While DLM are extremely nice to handle within
IDL.
> They are not the easiest to implement.
>
> Cheers,
> Haje
>
>
>
> "Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
> news:1078919071.346356@bengal...
>> Is there a simple guide to using call_external ? as i dont have the
> external
>> develpoment manual that the help page suggests.
>>
>> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>>
>> thanks
>>
>> Paul
>>

Subject: Re: Call_external
Posted by Haje Korth on Wed, 10 Mar 2004 17:20:50 GMT
View Forum Message <> Reply to Message

Paul,
the first thing you have to understand is that call_external is an advanced subject. Thus you will have to spend some time on the subject. It seems you are trying to auto-create the dll. To me this is even more advanced. Start simple: Take sum_array.c and compile it as dll. You need the switches -dll and -def:sum_array.def, where sum_array.def is the module definition file which contains your export. Study the compiler instruction on how to create the .def file. It will not work without it! Stick to the portable calling convention (if you actually open the .c file and read the information in the file you will understand what I mean). Once you create the dll, you can use sum_array.pro to test it. By using the method above you can separate problems in execution from problems in creating the dll. I never use autoglue, but from your errors below I am wondering whether IDL finds your compiler at all???

Haje

"Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message news:1078929414.412397@bengal...
> Thanks for that it looks easier than MS description of building shareable
> objects and def files. Only one small problem i'm having now is that
> all_callext_examples fails as it was unable to find some files. I assume its
> some problem so obvious its hitting me in the face, just cant see it.
>
> Anyone any ideas ?
>
> thanks
>
> IDL> all_callext_examples
> % Compiled module: ALL_CALLEXT_EXAMPLES.
> % Compiled module: SIMPLE_VARS.
> Calling simple_vars with the following arguments:
> B_L BYTE = 2
> I_L INT = 3
> L_L LONG = 4
> F_L FLOAT = 5.00000
> D_L DOUBLE = 6.0000000
> % Compiled module: GET_CALLEXT_EXLIB.
> The name specified is not recognized as an
> internal or external command, operable program or batch file.
> cl -D_DLL -DWIN32 -D_MT /nologo /I"E:\RSI\IDL60\external\include" /c
> "E:\RSI\IDL60\external\call_external\C\incr_struct.c"
> /Fo"incr_struct_166_PDPC.obj"

> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples_166_PDPC.exp
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples_166_PDPC.lib
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\incr_
> struct_166_PDPC.obj
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\simpl
> e_vars_166_PDPC.obj
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\strin
> g_array_166_PDPC.obj
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_a
> rray_166_PDPC.obj
> Could Not Find
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_2
> d_array_166_PDPC.obj
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: simple_vars, File =
>
C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples.dll
> ERROR_MOD_NOT_FOUND
> % Error occurred at: SIMPLE_VARS 91
> E:\RSI\IDL60\external\call_external\C\simple_vars.pro
> % ALL_CALLEXT_EXAMPLES 58
> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
> % $MAIN$
> % Execution halted at: ALL_CALLEXT_EXAMPLES 58
> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
>
> "Haje Korth" <haje.korth@jhuapl.edu> wrote in message
> news:c2n469$63v$1@aplcore.jhuapl.edu...
>> Paul,
>> Checkout the examples in the RSI\IDL60\external directory. They are
pretty
>> self-explanatory and are probably as descriptive as it will ever be. The

>> only book on external routines that I know of, was published by Ronn
Kling
>> and specializes on DLMs. While DLM are extremely nice to handle within
> IDL.
>> They are not the easiest to implement.
>>
>> Cheers,
>> Haje
>>
>>
>>
>> "Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
>> news:1078919071.346356@bengal...
>>> Is there a simple guide to using call_external ? as i dont have the
>> external
>>> develpoment manual that the help page suggests.
>>>
>>> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>>>
>>> thanks
>>>
>>> Paul
>>>
>
>
>

Subject: Re: Call_external
Posted by Karl Schultz on Wed, 10 Mar 2004 19:02:12 GMT
View Forum Message <> Reply to Message

It doesn't look like he's finding the compiler.

When you use the Visual Studio, the Studio finds and invokes the compiler for you. Outside the Studio, the compiler, linker, and other tools are not in your PATH, and so cannot be found if you try to invoke them from the command line or from IDL.

Look for a VSVARS32.BAT or a VCVARS32.BAT file in your directory where you installed Visual Studio. These BAT files modify/set the env vars PATH, INCLUDE, and LIB, which allows you to invoke the compiler tools from the command line. In order to use the compiler tools from IDL, you need to modify/set these environment variables using the System applet in the Windows Control Panel. The BAT file contains the values you need to set the env vars.

Karl

"Haje Korth" <haje.korth@jhuapl.edu> wrote in message
news:c2nipj$ol2$1@aplcore.jhuapl.edu...
> Paul,
> the first thing you have to understand is that call_external is an
advanced
> subject. Thus you will have to spend some time on the subject. It seems
you
> are trying to auto-create the dll. To me this is even more advanced. Start
> simple: Take sum_array.c and compile it as dll. You need the switches -dll
> and -def:sum_array.def, where sum_array.def is the module definition file
> which contains your export. Study the compiler instruction on how to
create
> the .def file. It will not work without it! Stick to the portable calling
> convention (if you actually open the .c file and read the information in
the
> file you will understand what I mean). Once you create the dll, you can
use
> sum_array.pro to test it. By using the method above you can separate
> problems in execution from problems in creating the dll. I never use
> autoglue, but from your errors below I am wondering whether IDL finds your
> compiler at all???
>
> Haje
>
>
> "Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
> news:1078929414.412397@bengal...
>> Thanks for that it looks easier than MS description of building
shareable
>> objects and def files. Only one small problem i'm having now is that
>> all_callext_examples fails as it was unable to find some files. I assume
> its
>> some problem so obvious its hitting me in the face, just cant see it.
>>
>> Anyone any ideas ?
>>
>> thanks
>>
>> IDL> all_callext_examples
>> % Compiled module: ALL_CALLEXT_EXAMPLES.
>> % Compiled module: SIMPLE_VARS.
>> Calling simple_vars with the following arguments:
>> B_L BYTE = 2
>> I_L INT = 3
>> L_L LONG = 4
>> F_L FLOAT = 5.00000
>> D_L DOUBLE = 6.0000000

>> % Compiled module: GET_CALLEXT_EXLIB.
>> The name specified is not recognized as an
>> internal or external command, operable program or batch file.
>> cl -D_DLL -DWIN32 -D_MT /nologo /I"E:\RSI\IDL60\external\include" /c
>> "E:\RSI\IDL60\external\call_external\C\incr_struct.c"
>> /Fo"incr_struct_166_PDPC.obj"
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
>> external_examples_166_PDPC.exp
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
>> external_examples_166_PDPC.lib
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\incr_
>> struct_166_PDPC.obj
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\simpl
>> e_vars_166_PDPC.obj
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\strin
>> g_array_166_PDPC.obj
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_a
>> rray_166_PDPC.obj
>> Could Not Find
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_2
>> d_array_166_PDPC.obj
>> % CALL_EXTERNAL: Error loading sharable executable.
>>   Symbol: simple_vars, File =
>>
>
 C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
>> external_examples.dll
>>   ERROR_MOD_NOT_FOUND

>> % Error occurred at: SIMPLE_VARS 91
>> E:\RSI\IDL60\external\call_external\C\simple_vars.pro
>> % ALL_CALLEXT_EXAMPLES 58
>> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
>> % $MAIN$
>> % Execution halted at: ALL_CALLEXT_EXAMPLES 58
>> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
>>
>> "Haje Korth" <haje.korth@jhuapl.edu> wrote in message
>> news:c2n469$63v$1@aplcore.jhuapl.edu...
>>> Paul,
>>> Checkout the examples in the RSI\IDL60\external directory. They are
> pretty
>>> self-explanatory and are probably as descriptive as it will ever be.
The
>>> only book on external routines that I know of, was published by Ronn
> Kling
>>> and specializes on DLMs. While DLM are extremely nice to handle within
>> IDL.
>>> They are not the easiest to implement.
>>>
>>> Cheers,
>>> Haje
>>>
>>>
>>>
>>> "Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
>>> news:1078919071.346356@bengal...
>>>> Is there a simple guide to using call_external ? as i dont have the
>>> external
>>>> develpoment manual that the help page suggests.
>>>>
>>>> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>>>>
>>>> thanks
>>>>
>>>> Paul
>>>>
>>
>>
>>
>
>

Subject: Re: Call_external
Posted by scuaic on Thu, 11 Mar 2004 22:32:46 GMT

Looks like your path variables are not set. You may want to reinstall
your MSVC++ and this time when it asks you to set your path variables
click yes. Using the VCVARS.BAT file does not help because IDL by
default open a new command line where the path variables are not set.
I know this because I had encountered the same problem when using
call_external.

Good Luck!

"Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
news:<1078929414.412397@bengal>...
> Thanks for that it looks easier than MS description of building shareable
> objects and def files. Only one small problem i'm having now is that
> all_callext_examples fails as it was unable to find some files. I assume its
> some problem so obvious its hitting me in the face, just cant see it.
>
> Anyone any ideas ?
>
> thanks
>
> IDL> all_callext_examples
> % Compiled module: ALL_CALLEXT_EXAMPLES.
> % Compiled module: SIMPLE_VARS.
> Calling simple_vars with the following arguments:
> B_L BYTE = 2
> I_L INT = 3
> L_L LONG = 4
> F_L FLOAT = 5.00000
> D_L DOUBLE = 6.0000000
> % Compiled module: GET_CALLEXT_EXLIB.
> The name specified is not recognized as an
> internal or external command, operable program or batch file.
> cl -D_DLL -DWIN32 -D_MT /nologo /I"E:\RSI\IDL60\external\include" /c
> "E:\RSI\IDL60\external\call_external\C\incr_struct.c"
> /Fo"incr_struct_166_PDPC.obj"
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples_166_PDPC.exp
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples_166_PDPC.lib
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\incr_
> struct_166_PDPC.obj
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\simpl
> e_vars_166_PDPC.obj

> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\strin
> g_array_166_PDPC.obj
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_a
> rray_166_PDPC.obj
> Could Not Find
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\sum_2
> d_array_166_PDPC.obj
> % CALL_EXTERNAL: Error loading sharable executable.
> Symbol: simple_vars, File =
> C:\WINNT\Profiles\ducksbury\.idl\idl_6_0_Win32_x86_m32_f64\c ompile_dir\call_
> external_examples.dll
> ERROR_MOD_NOT_FOUND
> % Error occurred at: SIMPLE_VARS 91
> E:\RSI\IDL60\external\call_external\C\simple_vars.pro
> % ALL_CALLEXT_EXAMPLES 58
> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
> % $MAIN$
> % Execution halted at: ALL_CALLEXT_EXAMPLES 58
> E:\RSI\IDL60\external\call_external\C\all_callext_examples.p ro
>
> "Haje Korth" <haje.korth@jhuapl.edu> wrote in message
> news:c2n469$63v$1@aplcore.jhuapl.edu...
>> Paul,
>> Checkout the examples in the RSI\IDL60\external directory. They are pretty
>> self-explanatory and are probably as descriptive as it will ever be. The
>> only book on external routines that I know of, was published by Ronn Kling
>> and specializes on DLMs. While DLM are extremely nice to handle within
> IDL.
>> They are not the easiest to implement.
>>
>> Cheers,
>> Haje
>>
>>
>>
>> "Dr Paul Ducksbury" <p.ducksbury@signal.qinetiq.com> wrote in message
>> news:1078919071.346356@bengal...
>>> Is there a simple guide to using call_external ? as i dont have the
> external
>>> develpoment manual that the help page suggests.
>>>
>>> Im using IDL 6.0 on PC (windows) with visual c++ 6.0
>>>
>>> thanks
>>>
>>> Paul

>>>

Subject: Re: call_external
Posted by Haje Korth on Sun, 17 Feb 2013 13:53:59 GMT
View Forum Message <> Reply to Message

On Friday, February 15, 2013 6:42:26 PM UTC-5, Katya wrote:
> Hello,
>
>
>
> I have an issue with call-external. When I try to use call_external, I receive the error "the program can't start because MSVCR90.dll is missing from your computer", but I have installed MVStudio2008, so I do have the msvcr90.dll on my computer. I realize that this is not purely IDL's question, but maybe someone could help me with the issue.
>
>
>
> Thanks!

Is the search path set correctly?

Subject: Re: call_external
Posted by Katerina Yakimenko on Thu, 07 Mar 2013 13:55:45 GMT
View Forum Message <> Reply to Message

On Sunday, February 17, 2013 3:53:59 PM UTC+2, Haje Korth wrote:
> On Friday, February 15, 2013 6:42:26 PM UTC-5, Katya wrote:
>
>> Hello,
>
>>
>
>>
>
>>
>
>> I have an issue with call-external. When I try to use call_external, I receive the error "the program can't start because MSVCR90.dll is missing from your computer", but I have installed MVStudio2008, so I do have the msvcr90.dll on my computer. I realize that this is not purely IDL's question, but maybe someone could help me with the issue.
>
>>
>
>>

>
>>
>
>> Thanks!
>
>
>
> Is the search path set correctly?

Oh, hello, I've just noticed your reply :-) Yes, path was set correctly, everything seemed ok. I eventually solved my problem with SPAWN and reading/writing from files which was awful ((. Anyway, thank you for the reply -- I've posted two questions to the group and up to now you are the only person who have replied)).

Subject: Re: call_external
Posted by David Fanning on Thu, 07 Mar 2013 14:10:55 GMT
View Forum Message <> Reply to Message

Katya writes:

> I've posted two questions to the group and up to now you are the only
> person who have replied).

Your questions are too hard. Do you have any easy ones? ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: call_external
Posted by Katerina Yakimenko on Thu, 07 Mar 2013 14:23:44 GMT
View Forum Message <> Reply to Message

On Thursday, March 7, 2013 4:10:55 PM UTC+2, David Fanning wrote:
> Katya writes:
>
>
>
>> I've posted two questions to the group and up to now you are the only

>
>> person who have replied).
>
>
>
> Your questions are too hard. Do you have any easy ones? ;-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

:-) Yes, but when I have easy questions I think you will laugh at me and sort them out on my own. If seriously, my Russian-language issue is quite vital since I have to insert pictures in my thesis strictly with Russian annotations and I writhe in horrow imaging how I will correct all annotations in a picture editor...
P.S.Thank you by the way for your book, it helped me greatly!

Katya.

Subject: Re: call_external
Posted by David Fanning on Thu, 07 Mar 2013 14:42:16 GMT
View Forum Message <> Reply to Message

Katya writes:

> If seriously, my Russian-language issue is quite vital since I have to insert pictures in my thesis strictly with Russian annotations and I writhe in horrow imaging how I will correct all annotations in a picture editor...

There used to be information about language catalogs in the IDL documentation, but I can find no mention of this is the current help

system that comes with IDL. This doesn't mean it's not there. It just means there is no way of finding it, except accidentally, while you are looking for something else. Maybe you have more pull with the documentation gods than I do. Wish I could be of more help.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")
