Subject: Re: call_external (IDL5.5)
Posted by R.Bauer on Mon, 25 Feb 2002 13:19:05 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:
>
Hi,
I give an example:

The following compiled and used by idl5.4.1 works fine and IDL5.5 got's
a memory fault.

IDL> PRINT,call_external('my_stat.so.linux','uid','file.txt',/i_v alue)

```
/*
Linux:
  gcc -fPIC -Wall -g -c my_stat.c
  gcc -g -shared -W1,-soname,my_stat.so.0 -o my_stat.so.linux my_stat.o
-lc
*/

#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>

typedef struct {
  unsigned short slen;
  short stype;
  char *s;
} IDL_STRING;

double uid(int argc, void *argv[])
{
  /* returns the user-id of the file */
  struct stat s;
  int rcode;
  IDL_STRING path;
  char buf[1023];
  long uid;


  path=*(IDL_STRING *) argv[0];
  strcpy(buf,path.s);
  buf[path.slen]='\0';

  rcode=stat(buf,&s);
```

```
  uid=s.st_uid;

  return(uid);

}
```

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
 ========== ======

Subject: Re: call_external (IDL5.5)
Posted by Gert Van de Wouwer on Mon, 25 Feb 2002 13:35:25 GMT
View Forum Message <> Reply to Message

I had the similar problem: a dll that worked under 5.4 didnt under 5.5.
recompile the dll while linking with the idl32.lib from the 5.5 distribution
solved it. This was off course under win32..

"Reimar Bauer" <r.bauer@fz-juelich.de> wrote in message
news:3C7A252F.F64A3633@fz-juelich.de...
> Hi,
>
> I have a found a difference between CALL_EXTERNAL for IDL5.4.1 and
> IDL5.5. One of my routines crashes in idl5.5.
> Why? I don't know at the moment!
>
> I don't find a hint about changes for call_external.
>
> What is your experience?
>

> Reimar
>
> --
> Reimar Bauer
>
> Institut fuer Stratosphaerische Chemie (ICG-1)
> Forschungszentrum Juelich
> email: R.Bauer@fz-juelich.de
> http://www.fz-juelich.de/icg/icg1/
> ========== ======
> a IDL library at ForschungsZentrum Juelich
> http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
>
> http://www.fz-juelich.de/zb/text/publikation/juel3786.html
> ========== ======

Subject: Re: call_external (IDL5.5)
Posted by notspecified on Mon, 25 Feb 2002 13:56:14 GMT
View Forum Message <> Reply to Message

On Mon, 25 Feb 2002 14:35:25 +0100, "Gert Van de Wouwer"
<Gert.VandeWouwer@NOSPAMua.ac.be> wrote:

> I had the similar problem: a dll that worked under 5.4 didnt under 5.5.
> recompile the dll while linking with the idl32.lib from the 5.5 distribution
> solved it. This was off course under win32..
>

Under what conditions do you have to link in idl32.lib? I don't
-think- I did that with my call_external .dll's under 5.4 (unless
there was something horrible buried in the export.h file...)

Matt Feinstein does not include his email address
in the text of usenet postings.

Harvard Law of Automotive Repair: Anything that goes away
by itself will come back by itself.

Subject: Re: call_external (IDL5.5)
Posted by Mark Rivers on Mon, 25 Feb 2002 14:56:51 GMT
View Forum Message <> Reply to Message

Reimar Bauer <r.bauer@fz-juelich.de> wrote in message
news:3C7A252F.F64A3633@fz-juelich.de...
> Hi,

>
> I have a found a difference between CALL_EXTERNAL for IDL5.4.1 and
> IDL5.5. One of my routines crashes in idl5.5.
> Why? I don't know at the moment!
>
> I don't find a hint about changes for call_external.
>
> What is your experience?

There is one very important difference between CALL_EXTERNAL in IDL 5.5 and previous versions, which has to do with how IDL strings are passed. The following code is from "export.h" in IDL 5.5.

```
*****************************************
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;
*****************************************
```

Note that "slen" is of type "int". In previous versions it was of type "short". Thus, if your external C code is being passed strings it needs to be changed for IDL 5.5.

I worked around this problem, making a single DLL that will work for any IDL version, by changing my IDL wrapper routines and C code to never pass strings, but convert everything to byte arrays before CALL_EXTERNAL, and back to strings after CALL_EXTERNAL.

Mark Rivers

Subject: Re: call_external (IDL5.5)
Posted by James Kuyper on Mon, 25 Feb 2002 15:28:49 GMT
View Forum Message <> Reply to Message

Mark Rivers wrote:
...

> There is one very important difference between CALL_EXTERNAL in IDL 5.5 and
> previous versions, which has to do with how IDL strings are passed. The
> following code is from "export.h" in IDL 5.5.
> ***
> 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;

Which points out he importance of using "#include export.h" to create
the typedef, rather than copying the typedef into your own code.

Subject: Re: call_external (IDL5.5)
Posted by Mark Rivers on Tue, 26 Feb 2002 04:03:24 GMT
View Forum Message <> Reply to Message

James Kuyper <kuyper@gscmail.gsfc.nasa.gov> wrote in message
news:3C7A5831.8030306@gscmail.gsfc.nasa.gov...
> Mark Rivers wrote:
> ...
>
>> There is one very important difference between CALL_EXTERNAL in IDL 5.5
and
>> previous versions, which has to do with how IDL strings are passed. The
>> following code is from "export.h" in IDL 5.5.
>> **
>> 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;
>
> Which points out he importance of using "#include export.h" to create
> the typedef, rather than copying the typedef into your own code.

This would not solve the problem that the original poster had, which is that
DLLs that work with IDL 5.4 no longer work with IDL 5.5. I want to
distribute a single DLL that will work with "all" IDL versions, and there is
no longer any way to do that if strings are passed to CALL_EXTERNAL. By
using byte arrays one can work around this problem.

Mark Rivers

Subject: Re: call_external (IDL5.5)
Posted by R.Bauer on Tue, 26 Feb 2002 16:58:51 GMT
View Forum Message <> Reply to Message

James Kuyper wrote:
>
> Mark Rivers wrote:
> ...
>
>> There is one very important difference between CALL_EXTERNAL in IDL 5.5 and
>> previous versions, which has to do with how IDL strings are passed. The
>> following code is from "export.h" in IDL 5.5.
>> **
>> 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;
>
> Which points out he importance of using "#include export.h" to create
> the typedef, rather than copying the typedef into your own code.

I agree to this and sure this was the main problem.

The second problem comes from this two lines.

```
strcpy(buf,path.s);
buf[path.slen]='\0';
```

I have learned that I have to do:

```
buf[path.slen]='\0';
strcpy(buf,path.s);
```

regards
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

http://www.fz-juelich.de/icg/icg1/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
 ========== ======

Subject: Re: call_external (IDL5.5)
Posted by Craig Markwardt on Tue, 26 Feb 2002 18:43:11 GMT
View Forum Message <> Reply to Message

Reimar Bauer <r.bauer@fz-juelich.de> writes:

> James Kuyper wrote:
...
>> Which points out he importance of using "#include export.h" to create
>> the typedef, rather than copying the typedef into your own code.
>
> I agree to this and sure this was the main problem.
>
> The second problem comes from this two lines.
>
> strcpy(buf,path.s);
> buf[path.slen]='\0';
>
> I have learned that I have to do:
>
> buf[path.slen]='\0';
> strcpy(buf,path.s);

Reimar, this does not sound right. The use of strcpy() is rather
unsafe, and you should prefer strncpy(). Also, there is the risk that
the "path" string is longer than buf[]. Then what do you do?

Safer approach:

```
char buf[NCHAR];
int len;

if (path.slen >= NCHAR) len = NCHAR-1; else len = path.slen;
if (len > 0) strncpy(buf, path.s, len);
buf[len] = 0;
```

If the "path" is too long for the buf[], then this code truncates it.

Cheers,

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 (IDL5.5)
Posted by Gert Van de Wouwer on Wed, 06 Mar 2002 14:42:02 GMT
View Forum Message <> Reply to Message

> Under what conditions do you have to link in idl32.lib? I don't
> -think- I did that with my call_external .dll's under 5.4 (unless
> there was something horrible buried in the export.h file...)
>

you only need to link if you use stuff like IDL_Message(..) in your
c-code...