Subject: Re: nested structures in dlm
Posted by Karl Schultz on Tue, 16 Jan 2007 21:16:26 GMT
View Forum Message <> Reply to Message

On Fri, 12 Jan 2007 04:55:50 -0800, lbusoni wrote:

> HI Guru's of DLMs,
>

snip

> It seems that me and IDL_MakeStruct got confused
> Any idea of what's happening? My code is completely crazy?
> Thanks
> Lorenzo

I compiled your code and it seemed to work fine for me.

```
** Structure FOO, 5 tags, length=400, data length=400:
   V000            STRUCT    -> V000 Array[1]
   V001            STRUCT    -> V001 Array[1]
   V002            STRUCT    -> V002 Array[1]
   V003            STRUCT    -> V003 Array[1]
   V004            STRUCT    -> V004 Array[1]
```

I did this on Windows with the C compiler, not C++. So I had to rearrange some variable declarations, but nothing that would change anything. I also did not supply idl_free_cb to IDL_ImportArray just because I was lazy, but that should not be the problem either.

So, I don't know what's wrong - it should work.

Karl

Subject: Re: nested structures in dlm
Posted by lbusoni on Wed, 17 Jan 2007 14:20:46 GMT
View Forum Message <> Reply to Message

Karl,
thanks for the reply.

Sometimes it works fine for me too.
Could you please try again increasing n_of_objects in order to maximize the probability of failure??
(with n_of_objects=250 I got a 10/10 of failures)

When it works fine, then it works fine forever in the current idl

session.
But if I stop and rerun IDL, the bad behaviuor can pop up again (sorry to be so generic, but I can't find a completely deterministic behaviour in this bug).

I tried to compile both with C and C++ compiler (on Linux). gcc is 4.1.2, idl is 6.2.

lbusoni$ gcc -Wall -shared -o tests.so wrapper_prova.cpp -I/usr/local/rsi/idl/external/include -lstdc++
lbusoni$ gcc -v
Using built-in specs.
Target: i486-linux-gnu
Configured with: ../src/configure -v --enable-languages=c,c++,fortran,objc,obj-c++,treelang --prefix=/usr --enable-shared --with-system-zlib --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --enable-nls --program-suffix=-4.1 --enable-__cxa_atexit --enable-clocale=gnu --enable-libstdcxx-debug --enable-mpfr --enable-checking=release i486-linux-gnu
Thread model: posix
gcc version 4.1.2 20060928 (prerelease) (Ubuntu 4.1.1-13ubuntu5)
lbusoni$ idl
IDL Version 6.2 (linux x86 m32). (c) 2005, Research Systems, Inc.

Lorenzo

Karl Schultz wrote:
> On Fri, 12 Jan 2007 04:55:50 -0800, lbusoni wrote:
>
>> HI Guru's of DLMs,
>>
>
> snip
>
>> It seems that me and IDL_MakeStruct got confused
>> Any idea of what's happening? My code is completely crazy?
>> Thanks
>> Lorenzo
>
> I compiled your code and it seemed to work fine for me.
>
> ** Structure FOO, 5 tags, length=400, data length=400:
>   V000   STRUCT   -> V000 Array[1]
>   V001   STRUCT   -> V001 Array[1]
>   V002   STRUCT   -> V002 Array[1]

> V003 STRUCT -> V003 Array[1]
> V004 STRUCT -> V004 Array[1]
>
> I did this on Windows with the C compiler, not C++. So I had to rearrange
> some variable declarations, but nothing that would change anything. I
> also did not supply idl_free_cb to IDL_ImportArray just because I was
> lazy, but that should not be the problem either.
>
> So, I don't know what's wrong - it should work.
>
> Karl

Subject: Re: nested structures in dlm
Posted by Karl Schultz on Thu, 18 Jan 2007 17:34:31 GMT
View Forum Message <> Reply to Message

On Wed, 17 Jan 2007 06:20:46 -0800, lbusoni wrote:

> Karl,
> thanks for the reply.
>
> Sometimes it works fine for me too.
> Could you please try again increasing n_of_objects in order to
> maximize the probability of failure??
> (with n_of_objects=250 I got a 10/10 of failures)
>
> When it works fine, then it works fine forever in the current idl
> session.
> But if I stop and rerun IDL, the bad behaviuor can pop up again (sorry
> to be so generic, but I can't find a completely deterministic behaviour
> in this bug).
>
> I tried to compile both with C and C++ compiler (on Linux). gcc is
> 4.1.2, idl is 6.2.
>
> lbusoni$ gcc -Wall -shared -o tests.so wrapper_prova.cpp
> -I/usr/local/rsi/idl/external/include -lstdc++
> lbusoni$ gcc -v
> Using built-in specs.
> Target: i486-linux-gnu
> Configured with: ../src/configure -v
> --enable-languages=c,c++,fortran,objc,obj-c++,treelang --prefix=/usr
> --enable-shared --with-system-zlib --libexecdir=/usr/lib
> --without-included-gettext --enable-threads=posix --enable-nls
> --program-suffix=-4.1 --enable-__cxa_atexit --enable-clocale=gnu
> --enable-libstdcxx-debug --enable-mpfr --enable-checking=release
> i486-linux-gnu

> Thread model: posix
> gcc version 4.1.2 20060928 (prerelease) (Ubuntu 4.1.1-13ubuntu5)
> lbusoni$ idl
> IDL Version 6.2 (linux x86 m32). (c) 2005, Research Systems, Inc.
>
> Lorenzo
>
>
>
> Karl Schultz wrote:
>> On Fri, 12 Jan 2007 04:55:50 -0800, lbusoni wrote:
>>
>>> HI Guru's of DLMs,
>>>
>>
>> snip
>>
>>> It seems that me and IDL_MakeStruct got confused
>>> Any idea of what's happening? My code is completely crazy?
>>> Thanks
>>> Lorenzo
>>
>> I compiled your code and it seemed to work fine for me.
>>
>> ** Structure FOO, 5 tags, length=400, data length=400:
>>   V000   STRUCT   -> V000 Array[1]
>>   V001   STRUCT   -> V001 Array[1]
>>   V002   STRUCT   -> V002 Array[1]
>>   V003   STRUCT   -> V003 Array[1]
>>   V004   STRUCT   -> V004 Array[1]
>>
>> I did this on Windows with the C compiler, not C++. So I had to rearrange
>> some variable declarations, but nothing that would change anything. I
>> also did not supply idl_free_cb to IDL_ImportArray just because I was
>> lazy, but that should not be the problem either.
>>
>> So, I don't know what's wrong - it should work.
>>
>> Karl

OK, I found the problem. You need to add a line of code, marked below:

```
   // I need to create the IDL_STRUCT_TAG_DEF [] at run time
   // because I don't know a priori the number of objects
   struct_tags = (IDL_STRUCT_TAG_DEF*)
malloc(sizeof(IDL_STRUCT_TAG_DEF) * (n_of_objects+1) );
   for (i=0; i<n_of_objects; i++){
```

```
      tag =  &struct_tags[i];
      tag->name=(char*)malloc(5);
      snprintf(tag->name,5,"V%03d",i);
      tag->dims=(IDL_MEMINT*) malloc(2*sizeof(IDL_MEMINT));
      tag->dims[0]=1;
      tag->dims[1]=1;
      tag->type=NULL;
      tag->flags = 0;   // NEW LINE
   }
```

The flags field is defined in the IDL_STRUCT_TAG_DEF struct in idl_exports.h

A lot of people write:

```
static IDL_STRUCT_TAG_DEF substruct_tags[] = {
   {"TIME",   times_dims, (void *) IDL_TYP_DOUBLE},
   {"FORCE",  force_dims, (void *) IDL_TYP_DOUBLE},
   {0}
 };
```

and forget to specify the flags member at the end. The C compiler fills it in as zero when you declare it statically like this. When you create struct tag defs dynamically, you must initialize this field.

Karl

Subject: Re: nested structures in dlm
Posted by lbusoni on Fri, 19 Jan 2007 09:36:36 GMT
View Forum Message <> Reply to Message

Karl,
thank you so much!
Now the dynamic definition of structures works like a charm!
Lorenzo
