Subject: Re: callable IDL and structures
Posted by David Fanning on Mon, 08 Apr 2002 18:42:19 GMT
View Forum Message <> Reply to Message

Sebastian Moeller (sebastian.moeller@lur.rwth-aachen.de) writes:

> at my place of work we have a C++ project which is to use IDL for user
> definable output. As we are somewhat bound to windows (but do not really
> like activeX) we decided that callable IDL might be the way to go. We
> intended to use structures to pass data to IDL. The strange thing is
> now, the structures we pass show their tags if we invoke "help,
> MyStructure ,/STRUCTURE". We see all tag names and types and content as
> defined in the C++ project. But every command that tries to access the
> data in the structure from IDL (e.g. tmp= MyStructure.MyTag1 with MyTag1
> is the first element of the structure) just tells that MyTag1 is not
> defined in MyStructure.

I'm afraid I'm skeptical about this. :-)

> Beeing challenged in that way we found that tmp= MyStructure.(n) with n
> beeing the "address" of the tag, actually gives us access to the data.
> "MyStructure.[0]" by the way does not work.

Yes, this is correct (on both counts). Ten minutes of looking
through on-line help doesn't come up with anything, but I know
it is documented *somewhere*. This is an alternative way to
access the fields of a structure. But if they are available
this way, they are sure as heck available via the fields names
as well. I'd like to see the code you are using to "access" them
by their names.

> But as this behavior is not
> documented anywhere (well, at least not where we searched) we assume it
> is rather daring to go relay on the stability of this feature for the
> future. (Heck, RSI even changed the behavior of SIZE() between 5.3 and
> 5.4, without mentioning in the what's new in 5.4. So maybe stability is
> relative anyhow ;)...)

I think you are fine doing this.

> So after too long a story, is there anybody out there who knows where we
> went wrong or whether this might be considered a bug worth reporting to RSI?

I don't see any bugs...yet.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: callable IDL and structures
Posted by Sebastian Moeller on Mon, 08 Apr 2002 20:13:44 GMT
View Forum Message <> Reply to Message

Hello Mr. Fanning

this really is a nice and impressivly quick news group. Thank you very much for answering.

David Fanning wrote:
[schnipp]
> Yes, this is correct (on both counts). Ten minutes of looking
> through on-line help doesn't come up with anything, but I know
> it is documented *somewhere*. This is an alternative way to
> access the fields of a structure. But if they are available
> this way, they are sure as heck available via the fields names
> as well. I'd like to see the code you are using to "access" them
> by their names.
[schnapp]
OK, here we go... This is the part in the C++ project. It is supposed to define the named structure str

```
typedef
struct
{
 int testarr[50];
 int field2;
} teststr;

IDL_MEMINT d1[10];
d1[0]=1;
d1[1]=50;
IDL_STRUCT_TAG_DEF sd[4];
sd[0].name="testarr";
sd[0].type=(void*)IDL_TYP_LONG;
sd[0].dims=(IDL_MEMINT*)d1;
sd[0].flags=NULL;
sd[1].name="field2";
sd[1].type=(void*)IDL_TYP_LONG;
sd[1].dims=NULL;
```

```
  sd[1].flags=NULL;
  sd[2].name=NULL;
  sd[2].type=NULL;
  sd[2].dims=NULL;
  sd[2].flags=NULL;


  IDL_MEMINT d[3];
  d[0]=1;
  for(int k=0; k<50; k++)
  {
   teststr.testarr[k]=k;
  }
  teststr.field2=10;
  void* ss=IDL_MakeStruct("str",sd);



var=IDL_ImportNamedArray("teststr",1,d,IDL_TYP_STRUCT,(UCHAR*)(&teststr),0,ss);

  int err=IDL_ExecuteStr("help, /structure, teststr");

/* this line gives the whole information for the structure, including
the tag names and tag types as defined in this c fragment (teststarr and
field2). Also the data contained in the structure (the array containing
the subscript values and 10 for field2) is shown correctly.*/

  err=IDL_ExecuteStr("print, teststr.field2");

/*this line gives an error message about field2 not beeing defined in
teststr, even thoug the preceedind command showed that teststr contained
the tags testarr and field2*/

  err=IDL_ExecuteStr("print, teststr.(where(TAG_NAMES(teststr) EQ 'testarr'))");

/*this line actually works, it prints the content of the array (numbers
from 0 to 49). It is a rude attempt to bring the structure tag names
back into the game, and a pretty obfusceted way to write "print,
teststr.(0)". A few defines along the lines of: #define testarr (0),
would save the day...*/

 To sum it up we are able to pass the data to IDL. And IDL has all the
meta-information about the structure's contents (tags, types and data).
The tag names can even be seen from the TAG_NAMES function, but attempts
to address the tags directly fail.
 I hope this information serves to illustrate the problem more deeply.

Ahoi
```

Sebastian Moeller

Subject: Re: callable IDL and structures
Posted by rigby on Mon, 08 Apr 2002 21:46:35 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.171b92747f3d317a989876@news.frii.com>...
> Sebastian Moeller (sebastian.moeller@lur.rwth-aachen.de) writes:
>

>> Beeing challenged in that way we found that tmp= MyStructure.(n) with n
>> beeing the "address" of the tag, actually gives us access to the data.
>> "MyStructure.[0]" by the way does not work.

> Yes, this is correct (on both counts). Ten minutes of looking
> through on-line help doesn't come up with anything, but I know
> it is documented *somewhere*.
> David

It's in the section "Advanced Structure Usage" of the online help.
[I remember this only because I had to spend considerably more than
ten minutes one time trying to find it. :)]

--Wayne

Subject: Re: callable IDL and structures
Posted by Sebastian Moeller on Tue, 09 Apr 2002 08:00:48 GMT
View Forum Message <> Reply to Message

Hello Mr. Fanning

this really is a nice and impressivly quick news group. Thank you very
much for answering.

David Fanning wrote:
[schnipp]
Yes, this is correct (on both counts). Ten minutes of looking through
on-line help doesn't come up with anything, but I know
it is documented *somewhere*. This is an alternative way to
access the fields of a structure. But if they are available
this way, they are sure as heck available via the fields names
as well. I'd like to see the code you are using to "access" them
by their names.
[schnapp]

OK, here we go... This is the part in the C++ project. It is supposed to
define the named structure str

```
typedef
struct
{
   int testarr[50];
   int field2;
} teststr;

IDL_MEMINT d1[10];
d1[0]=1;
d1[1]=50;
IDL_STRUCT_TAG_DEF sd[4];
sd[0].name="testarr";
sd[0].type=(void*)IDL_TYP_LONG;
sd[0].dims=(IDL_MEMINT*)d1;
sd[0].flags=NULL;
sd[1].name="field2";
sd[1].type=(void*)IDL_TYP_LONG;
sd[1].dims=NULL;
sd[1].flags=NULL;
sd[2].name=NULL;
sd[2].type=NULL;
sd[2].dims=NULL;
sd[2].flags=NULL;


IDL_MEMINT d[3];
d[0]=1;
for(int k=0; k<50; k++)
{
   teststr.testarr[k]=k;
}
teststr.field2=10;
void* ss=IDL_MakeStruct("str",sd);



var=IDL_ImportNamedArray("teststr",1,d,IDL_TYP_STRUCT,(UCHAR*)(&teststr),0,ss);

int err=IDL_ExecuteStr("help, /structure, teststr");

/* this line gives the whole information for the structure, including
the tag names and tag types as defined in this c fragment (teststarr and
field2). Also the data contained in the structure (the array containing
the subscript values and 10 for field2) is shown correctly.*/

err=IDL_ExecuteStr("print, teststr.field2");
```

/*this line gives an error message about field2 not beeing defined in
teststr, even thoug the preceedind command showed that teststr contained
the tags testarr and field2*/

err=IDL_ExecuteStr("print, teststr.(where(TAG_NAMES(teststr) EQ
'testarr'))");

/*this line actually works, it prints the content of the array (numbers
from 0 to 49). It is a rude attempt to bring the structure tag names
back into the game, and a pretty obfusceted way to write "print,
teststr.(0)". A few defines along the lines of: #define testarr (0),
would save the day...*/

To sum it up we are able to pass the data to IDL. And IDL has all
the meta-information about the structure's contents (tags, types and
data). The tag names can even be seen from the TAG_NAMES function, but
attempts to address the tags directly fail.
I hope this information serves to illustrate the problem more deeply.

Ahoi
Sebastian Moeller

P.S.: Thios is my second attempt to post the reply, as the first seems
to be lost somewhere...

Subject: Re: callable IDL and structures
Posted by Stein Vidar Hagfors H[1] on Tue, 09 Apr 2002 15:33:23 GMT
View Forum Message <> Reply to Message

Sebastian Moeller <sebastian.moeller@lur.rwth-aachen.de> writes:

> Hi there IDL-experts,
>
> at my place of work we have a C++ project which is to use IDL for user
> definable output. As we are somewhat bound to windows (but do not
> really like activeX) we decided that callable IDL might be the way to
> go. We intended to use structures to pass data to IDL. The strange
> thing is now, the structures we pass show their tags if we invoke
> "help, MyStructure ,/STRUCTURE". We see all tag names and types and
> content as defined in the C++ project. But every command that tries to
> access the data in the structure from IDL (e.g. tmp=
> MyStructure.MyTag1 with MyTag1 is the first element of the structure)
> just tells that MyTag1 is not defined in MyStructure.

Just a hunch, based on your C++ background & writing things like "MyStructure.MyTag1" etc: Try all upper (or all lower) case for the structure tag names when *defining* them. My suspicion is that they're converted to one of those in the "parsing" part of IDL, but since your C++ code presumably short-circuits that and defines the structures by calling the appropriate function directly, this might go wrong.

--

Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
Mail Code 682.3, Bld. 26, Room G-1, Tel.: 1-301-286-9028/240-354-6066
Greenbelt, Maryland 20771, USA. Fax: 1-301-286-0264

Subject: Re: callable IDL and structures
Posted by Sebastian Moeller on Thu, 11 Apr 2002 11:25:54 GMT
View Forum Message <> Reply to Message

Hi

Thank you very much David, wayne and Stein. All of your comments have been very helpful. It took some time though to get back to the C++ machine but now things work likke they are supposed to be. While it was assuring to find that structure.(tag_address) actually is documented, what really saved the day is the fact that the tag names had to be declarded ALL UPPERCASE in the C++ project. Since we done that things work fine.

Ahoi & thanks a load

Sebastian Moeller
