Subject: dynamic memory in dll
Posted by Gert Van de Wouwer on Mon, 03 Jun 2002 15:21:49 GMT
View Forum Message <> Reply to Message

Hi,

I want to use C++ code through a dll with call_external. Is it possible to allocate IDL memory for a variable that is alive after the dll unloads? I mean:

a= PTR_NEW(/ALLOCATE_HEAP) (???)
call external (name, function, a, b, c,d ,...)
a now contains something (not fixed size)

Is this possible without going through the hassle of learning to cope with DLM's?

thx

Gert

Subject: repost: Re: dynamic memory in dll
Posted by Gert on Mon, 10 Jun 2002 07:52:33 GMT
View Forum Message <> Reply to Message

Thanks for the input Randall and Ronn.

Understandebly, noone's a fan of call_external. I used it for my previous project, but now I think I'll use linkimage. DLM's are next - i'm moving up... (I'll have to get Ronn's book).
Unfortunately, I have to get something finished in 2 weeks - no time to investigate too much further.

Now, could you comment on the code i put below?

This code seems to work with

a=float(indgen(5))
b=testdlm1(a)
print, b

I have 2 questions
1) Can I change the dimension a in the C-code
2) how can I use IDL_cvtflt if a is an int array? if i use float* pflTemp = (float*)IDL_CvtFlt(1, idlIn1) then pflTemp apparantly is a new temporary variable that has nothing to do with a - a is thus unchanged when returning to IDL.

...or am i just asking to much here?

thanks for the replies,

Gert

C-Code

```
IDL_VPTR TestDLM1(int argc, IDL_VPTR argv[])
{
int i;

//get first input
IDL_VPTR idlIn1 = argv[0];
IDL_ENSURE_SIMPLE(idlIn1);
IDL_ENSURE_ARRAY(idlIn1);
if (idlIn1->type != IDL_TYP_FLOAT)
    IDL_Message(IDL_M_GENERIC, IDL_MSG_LONGJMP, "Error! float array
expected");

float* pflIn = (float *)idlIn1->value.arr->data;
int dN = idlIn1->value.arr->n_elts;

for(i=0;i<dN;i++) pflIn[i] = pflIn[i] * (float)2.0;


//make a return vector
IDL_VPTR dst;
float* pflTemp =
 (float*)IDL_MakeTempVector(IDL_TYP_FLOAT,10,IDL_ARR_INI_INDE X, &dst);
pflTemp[0]=100.0;
pflTemp[3]=200.0;

return(dst);
}
```

*****end of code **********

"ronn kling" <ronn@rlkling.com> wrote in message
news:B9218AAC.545B%ronn@rlkling.com...
> in article 3cfb8997.0@news.ruca.ua.ac.be, Gert Van de Wouwer at

> Gert.VandeWouwer@NOSPAMua.ac.be wrote on 6/3/02 11:21 AM:
>
>> Hi,
>>
>> I want to use C++ code through a dll with call_external. Is it possible
to
>> allocate IDL memory for a variable that is alive after the dll unloads?
I
>> mean:
>>
> Hello Gert,
>
> Do you really want to pass in a pointer or do you just want to create an
> array in the dll and pass it back to IDL? i.e.
>
> ;a does not exist yet
> call external (name, function, a, b, c,d ,...)
> a now contains something (not fixed size)
>
> I avoid call_external at all costs. It is so much easier to write your
own
> dlm/dll even if you have to call another dll. If you choose this route
then
> you want to use the IDL_MakeTempArray in your dll and pass back the
IDL_VPTR
> it creates. That way you can use it in your IDL code just like a normal
> variable.
>
> -Ronn
>
>
> --
> Ronn Kling
> KRS, inc.
> email: ronn@rlkling.com
> "Application Development with IDL" programming book updated for IDL5.5!
> "Calling C from IDL, Using DLM's to extend your IDL code"
> http://www.rlkling.com/
>
>
>