
Subject: Re: dlm question

Posted by [David Shadovitz](#) on Tue, 12 Nov 2002 22:22:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm still waiting for David Fanning to reply that "there's no such thing as a DLM question."

-David Shadovitz

Eric Fournier wrote:

> I have a DLM (Dynamically Loadable Modules) question.
>
> I have create a dlm with system routine in the past, but does anyone know
> how to create
> objects and associated methods. I have looked into the idl_dataminer.dlm and
> it
> declares:
>
> MODULE DATAMINER
> DESCRIPTION IDL DATAMINER support
> VERSION 1.0.1
> SOURCE Research Systems, Inc.
> BUILD_DATE AUG 28 2001
> CHECKSUM 378MWGKG
> FUNCTION DB_EXISTS 0 0
> FUNCTION DIALOG_DBCONNECT 1 1 KEYWORDS
> FUNCTION IDLDBDATABASE::GETDATASOURCES 1 1
> FUNCTION IDLDBDATABASE::GETTABLES 1 1
> FUNCTION IDLDBDATABASE::INIT 1 1
>
> I can see that DB_EXISTS a a standard function, but IDLDBDATABASE is an
> object on wich you can
> call obj_new('IDLDBDATABASE') in IDL. Does anyone know how to code such
> thing in the dll associated in the .dlm file.
>
> I know some mecanism exist, IDL_SYSFUN_DEF_F_METHOD is mentionned in the
> export.h.
>
> Any help would be greatly appreciated
>
> Eric Fournier

Subject: Re: dlm question

Posted by [David Fanning](#) on Tue, 12 Nov 2002 22:51:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Shadovitz (david_shadovitz@xontech.com) writes:

> I'm still waiting for David Fanning to reply that "there's no such thing as a
> DLM question."

Well, I was going to say there is "no such thing as a DLM answer", but I guess that's obvious by now. :-)

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
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: dlm question
Posted by [Rick Towler](#) on Wed, 13 Nov 2002 00:47:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

I asked this question (in a more obtuse form) a couple of years ago and received no response. I understand the desire to do this although I would think that for most applications you could just wrap your dlm routines within a standard IDL object. This approach certainly isn't as cool as writing your object as a .dlm but I would guess that your time could be better spent elsewhere.

But since you piqued my interest again, and your .dlm file gave me an idea, this is what I have come up with.

First I think we should start with a simple c++ class. Here is the specification file "test.h":

```
#include "export.h"

class Test
{
public:
  IDL_VPTR IDL_CDECL Init(int argc, IDL_VPTR *argv);
private:
  long aprop;
  long bprop;
};
```

Just one public member, Init, and 2 private data members.

Next, we need an implementation file.

```
#define STRICT
#include <stdio.h>
#include <windows.h>
#include "test.h"
#include "export.h"

#define ARRLEN(arr) (sizeof(arr)/sizeof(arr[0]))

/* Test object Init function*/
IDL_VPTR IDL_CDECL Test::Init(int argc, IDL_VPTR *argv)
{

    IDL_ENSURE_SCALAR(argv[0]);
    IDL_ENSURE_SCALAR(argv[1]);

    aprop = IDL_LongScalar(argv[0]);
    bprop = IDL_LongScalar(argv[1]);

    return IDL_GettmpLong(1);

}

int IDL_Load(void)
{

    static IDL_SYSFUN_DEF2 function_addr[] = {
        {(IDL_SYSRtn_GENERIC) Test::Init, "TEST::INIT", 2, 2,
        IDL_SYSFUN_DEF_F_METHOD, 0},
    };

    return IDL_SysRtnAdd(function_addr, TRUE, ARRLEN(function_addr));
}
```

You will note I have used `IDL_SYSFUN_DEF_F_METHOD` in the `IDL_SYSFUN_DEF2` struct definition. This pretty much has all been a giant guess based on my limited experience with `dlms/c/c++`.

But.

I can't compile it. There is a problem with the `funct_addr` aspect of the `IDL_SYSFUN_DEF2` struct. Well, I don't know exactly where the problem is but the compiler chokes if the function address is in a `class::member` form. The error is:

```
objectTest.cpp(99) : error C2440: 'type cast' : cannot convert from '' to  
'IDL_VARIABLE *(__cdecl*)(void)'  
None of the functions with this name in scope match the target type
```

Any ideas?

-Rick

"Eric Fournier" <eric.fournier@videotron.ca> wrote

- > I have a DLM (Dynamically Loadable Modules) question.
- >
- > I have create a dlm with system routine in the past, but does anyone know
- > how to create
- > objects and associated methods. I have looked into the `idl_dataminer.dlm`
- > and
- > it
- > declares:
- >
- > MODULE DATAMINER
- > DESCRIPTION IDL DATAMINER support
- > VERSION 1.0.1
- > SOURCE Research Systems, Inc.
- > BUILD_DATE AUG 28 2001
- > CHECKSUM 378MWGKG
- > FUNCTION DB_EXISTS 0 0
- > FUNCTION DIALOG_DBCONNECT 1 1 KEYWORDS
- > FUNCTION IDLDBDATABASE::GETDATASOURCES 1 1
- > FUNCTION IDLDBDATABASE::GETTABLES 1 1
- > FUNCTION IDLDBDATABASE::INIT 1 1
- >
- > I can see that `DB_EXISTS` a a standard function, but `IDLDBDATABASE` is an
- > object on wich you can
- > call `obj_new('IDLDBDATABASE')` in IDL. Does anyone know how to code such
- > thing in the dll associated in the `.dlm` file.
- >
- > I know some mecanism exist, `IDL_SYSFUN_DEF_F_METHOD` is mentionned in the
- > `export.h`.

>
> Any help would be greatly appreciated
>
>
> Eric Fournier
>
>
>

Subject: Re: dlm question
Posted by [ronn](#) on Wed, 13 Nov 2002 13:22:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Eric,

The short answer is that RSI hasn't released the information on how to do this.

The long answer is that with a little hacking you can figure out how to make a method call a DLM, but not creating an IDL object. But the "self" structure that gets passed to the C dlm method is not of much use. That is you can't use it to call other methods and so on.

It has been while since I looked at this and you have made me curious again. I am going to look at some old code that I have and if I come up with anything I will let you know.

-Ronn

--
Ronn Kling
KRS, inc.
email: ronn@rlkling.com
"Application Development with IDL" ½ programming book updated for IDL5.5!
"Calling C from IDL, Using DLM's to extend your IDL code!"
"Power Graphics with IDL, A Beginner's Guide to Object Graphics", NEW BOOK!
<http://www.rlkling.com/>

in article [PsaA9.5365\\$Vr1.772164@news20.bellglobal.com](mailto:PsaA9.5365$Vr1.772164@news20.bellglobal.com), Eric Fournier at eric.fournier@videotron.ca wrote on 11/12/02 12:00 PM:

> I have a DLM (Dynamically Loadable Modules) question.
>

Subject: Re: dlm question

Posted by [Randy Frank](#) on Thu, 21 Nov 2002 03:52:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Eric,

Consider the IDL naming scheme for objects and consider the mechanism for calling a C++ method from C in something like VC++ (take DirectX as an example). This should give you some idea. Ronn is correct in that RSI has not released this information and there is no direct way of doing this. That having been said, with some creativity, one can write some accessor functions that will get you 90% of the way there, along the lines of what Ronn suggests (playing with the 'self' reference creatively) until RSI exposes the necessary functions. I'm not sure what this buys you over a object written as a .PRO file that happens to call out to some 'C' functions in a DLM however (an approach that will not break over time)...

Thanks.

ronn kling wrote:

> Hi Eric,

>

> The short answer is that RSI hasn't released the information on how to do
> this.

>

> The long answer is that with a little hacking you can figure out how to make
> a method call a DLM, but not creating an IDL object. But the "self"
> structure that gets passed to the C dlm method is not of much use. That is
> you can't use it to call other methods and so on.

>

> It has been while since I looked at this and you have made me curious again.
> I am going to look at some old code that I have and if I come up with
> anything I will let you know.

>

> -Ronn

>

>

> --

> Ronn Kling

> KRS, inc.

> email: ronn@rkl Kling.com

> "Application Development with IDL" programming book updated for IDL5.5!

> "Calling C from IDL, Using DLM's to extend your IDL code"!

> "Power Graphics with IDL, A Beginner's Guide to Object Graphics", NEW BOOK!

> <http://www.rkl Kling.com/>

>

>

>

> in article PsaA9.5365\$Vr1.772164@news20.bellglobal.com, Eric Fournier at
> eric.fournier@videotron.ca wrote on 11/12/02 12:00 PM:
>
>
>> I have a DLM (Dynamically Loadable Modules) question.
>>
>
>
>

Subject: Re: dlm question

Posted by [Randall Skelton](#) on Thu, 21 Nov 2002 13:34:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, 21 Nov 2002, Randall Frank wrote:

> Consider the IDL naming scheme for objects and consider the
> mechanism for calling a C++ method from C in something like VC++
> (take DirectX as an example). This should give you some idea.
> Ronn is correct in that RSI has not released this information and
> there is no direct way of doing this. That having been said,
> with some creativity, one can write some accessor functions that
> will get you 90% of the way there, along the lines of what Ronn
> suggests (playing with the 'self' reference creatively) until RSI
> exposes the necessary functions. I'm not sure what this buys you
> over a object written as a .PRO file that happens to call out to
> some 'C' functions in a DLM however (an approach that will not
> break over time)...

I tend to agree-- I've been writing IDL object code that calls my C dlms for the past year now. Last year I begged and pleaded with RSI to release this interface (even with a non-disclosure agreement) and had no luck getting it. I even did my bit to reverse engineer the interface which was quite successful. I would post what I learned but fear Craig-like repercussions. My main comment is that while gaining access to the object API would be nice, the holly grail would involve stable, direct access (via a C API) to the IDL heap variable. This, along with my request for proper operator over-loading in IDL objects are my top two requests. Are any of the lurkers at RSI listening?

Note to DF: Are you ready to compile a new top 10 list?

Cheers,
Randall
