
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_SYSRN_GENERIC) Test::Init, "TEST::INIT", 2, 2,
IDL_SYSFUN_DEF_F_METHOD, 0},
};

return IDL_SysRtnAdd(function_addr, TRUE, ARLEN(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
>
>
>
