
Subject: Re: How to call an IDL procedure from a DLM
Posted by [Haje Korth](#) on Thu, 23 Feb 2006 13:26:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Maurizio,

What you are trying to do is an interesting concecpt, but I don't think DLMs are intended to work this way. At least I have never run across an example for such use. Isn't the whole purpose of DLMs, `call_external`, etc. to call code that is specifically NOT implemeneted in IDL? Why would you bother writing a DLM if the functionality is already present in IDL?

My guess is that you probably have to go out and find an external C or FORTRAN library that does what you want, e.g. Numerical Recipes. RSI manuals often specify the source of their external algorithms. Or, if the algortihm is written in IDL, you can just translate it.

This may just be well above my head, so maybe Karl has some input on this. If this should really work, I would be interested in seeing an example. I would be afraid that each of these calls need to start an independent IDL session, thus leading to significant slowdown.

Cheers,
Haje

"Maurizio Tomasi" <tomasi@lambrate.inaf.it> wrote in message
news:[dthqki\\$eh1\\$1@muffin.area.ba.cnr.it](mailto:dthqki$eh1$1@muffin.area.ba.cnr.it)...

> Last week I posted a request for help, since my DLM produced a number of
> "Stopped on unknown instruction" errors. After some debugging, I found
> that the problem resides in `IDL_ExecuteStr`, which is not usable in DLMs
> since it has been meant to be used only with Callable IDL.
>
> However, I still need to call an IDL procedure from my DLM. I looked for a
> `IDL_CallProc` function, but I have had no success. Does somebody know if
> there is a way to do this?
>
> Thank you very much,
> Maurizio Tomasi.

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Rick Towler](#) on Thu, 23 Feb 2006 16:24:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Maurizio Tomasi wrote:

> Last week I posted a request for help, since my DLM produced a number of
> "Stopped on unknown instruction" errors. After some debugging, I found
> that the problem resides in `IDL_ExecuteStr`, which is not usable in DLMs
> since it has been meant to be used only with Callable IDL.

>
> However, I still need to call an IDL procedure from my DLM. I looked for
> a IDL_CallProc function, but I have had no success. Does somebody know
> if there is a way to do this?

I don't know if there is a way to do this but why do it the hard way?
Why don't you wrap your dlm's functions in an IDL function or object and
call the IDL procedures from within IDL?

-Rick?

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Karl Schultz](#) on Thu, 23 Feb 2006 16:47:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 23 Feb 2006 08:26:54 -0500, Haje Korth wrote:

> Maurizio,
> What you are trying to do is an interesting concecpt, but I don't think DLMs
> are intended to work this way. At least I have never run across an example
> for such use. Isn't the whole purpose of DLMs, call_external, etc. to call
> code that is specifically NOT implemeneted in IDL? Why would you bother
> writing a DLM if the functionality is already present in IDL?

I'm thinking that he is doing part of his DLM processing in C, and then
gets to a point where it would be Really Nice to call an IDL routine to
finish it up because the IDL routine is at-hand and might be a pain to
translate into C or whatever.

It should be possible to refactor the code though to do some things in C
and then return the intermediate results to IDL, where he can then call
the IDL code. I can't speculate further without knowing more about the
problem.

> My guess is that you probably have to go out and find an external C or
> FORTRAN library that does what you want, e.g. Numerical Recipes. RSI manuals
> often specify the source of their external algorithms. Or, if the algortihm
> is written in IDL, you can just translate it.
>
> This may just be well above my head, so maybe Karl has some input on
> this. If this should really work, I would be interested in seeing an
> example. I would be afraid that each of these calls need to start an
> independent IDL session, thus leading to significant slowdown.

You are on the right track. It isn't so much starting a new session, but
rather reentering the interpreter that is causing the problem.

I think redesigning the DLM as hinted above is the best approach. One way to "call IDL" from a DLM would be to spawn a new IDL process. But that would involve passing data back and forth via files or pipes/sockets and I think that's going to be more involved than refactoring the DLM.

Karl

>
> Cheers,
> Haje
>
> "Maurizio Tomasi" <tomasi@lambrate.inaf.it> wrote in message
> news:dthqki\$eh1\$1@muffin.area.ba.cnr.it...
>> Last week I posted a request for help, since my DLM produced a number
>> of "Stopped on unknown instruction" errors. After some debugging, I
>> found that the problem resides in IDL_ExecuteStr, which is not usable
>> in DLMs since it has been meant to be used only with Callable IDL.
>>
>> However, I still need to call an IDL procedure from my DLM. I looked
>> for a IDL_CallProc function, but I have had no success. Does somebody
>> know if there is a way to do this?
>>
>> Thank you very much,
>> Maurizio Tomasi.

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Haje Korth](#) on Thu, 23 Feb 2006 17:14:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick,
I just looked at this as a prove of concept, not as something I would try
really hard to make use of. Life is hard enough as it is! :-)

Haje

"Rick Towler" <rick.towler@nomail.noaa.gov> wrote in message
news:dtkps4\$jop\$1@news.nems.noaa.gov...
>
> Maurizio Tomasi wrote:
>> Last week I posted a request for help, since my DLM produced a number of
>> "Stopped on unknown instruction" errors. After some debugging, I found
>> that the problem resides in IDL_ExecuteStr, which is not usable in DLMs
>> since it has been meant to be used only with Callable IDL.
>>
>> However, I still need to call an IDL procedure from my DLM. I looked for

>> a IDL_CallProc function, but I have had no success. Does somebody know if
>> there is a way to do this?
>
> I don't know if there is a way to do this but why do it the hard way? Why
> don't you wrap your dlm's functions in an IDL function or object and call
> the IDL procedures from within IDL?
>
> -Rick?

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Haje Korth](#) on Thu, 23 Feb 2006 17:17:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Karl,
thanks for the info. So it seems even you folks at RSI think that this is
unfeasible. I usually blame things first on my C-illiteracy... :-)

"Karl Schultz" <k____schultz@rsinc.com> wrote in message
news:pan.2006.02.23.16.47.51.78000@rsinc.com...
> On Thu, 23 Feb 2006 08:26:54 -0500, Haje Korth wrote:
>
>> Maurizio,
>> What you are trying to do is an interesting concecpt, but I don't think
>> DLMS
>> are intended to work this way. At least I have never run across an
>> example
>> for such use. Isn't the whole purpose of DLMS, call_external, etc. to
>> call
>> code that is specifically NOT implemeneted in IDL? Why would you bother
>> writing a DLM if the functionality is already present in IDL?
>
> I'm thinking that he is doing part of his DLM processing in C, and then
> gets to a point where it would be Really Nice to call an IDL routine to
> finish it up because the IDL routine is at-hand and might be a pain to
> translate into C or whatever.
>
> It should be possible to refactor the code though to do some things in C
> and then return the intermediate results to IDL, where he can then call
> the IDL code. I can't speculate further without knowing more about the
> problem.
>
>> My guess is that you probably have to go out and find an external C or
>> FORTRAN library that does what you want, e.g. Numerical Recipes. RSI
>> manuals
>> often specify the source of their external algorithms. Or, if the
>> algortihm

>> is written in IDL, you can just translate it.
>>
>> This may just be well above my head, so maybe Karl has some input on
>> this. If this should really work, I would be interested in seeing an
>> example. I would be afraid that each of these calls need to start an
>> independent IDL session, thus leading to significant slowdown.
>
> You are on the right track. It isn't so much starting a new session, but
> rather reentering the interpreter that is causing the problem.
>
> I think redesigning the DLM as hinted above is the best approach. One way
> to "call IDL" from a DLM would be to spawn a new IDL process. But that
> would involve passing data back and forth via files or pipes/sockets and I
> think that's going to be more involved than refactoring the DLM.
>
> Karl
>
>
>>
>> Cheers,
>> Haje
>>
>> "Maurizio Tomasi" <tomasi@lambrate.inaf.it> wrote in message
>> news:dtqki\$eh1\$1@muffin.area.ba.cnr.it...
>>> Last week I posted a request for help, since my DLM produced a number
>>> of "Stopped on unknown instruction" errors. After some debugging, I
>>> found that the problem resides in IDL_ExecuteStr, which is not usable
>>> in DLMs since it has been meant to be used only with Callable IDL.
>>>
>>> However, I still need to call an IDL procedure from my DLM. I looked
>>> for a IDL_CallProc function, but I have had no success. Does somebody
>>> know if there is a way to do this?
>>>
>>> Thank you very much,
>>> Maurizio Tomasi.
>

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Maurizio Tomasi](#) on Fri, 24 Feb 2006 08:27:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Karl Schultz wrote:

> On Thu, 23 Feb 2006 08:26:54 -0500, Haje Korth wrote:
>
>> Maurizio,
>> What you are trying to do is an interesting concecpt, but I don't think DLMs
>> are intended to work this way. At least I have never run across an example

>> for such use. Isn't the whole purpose of DLMs, call_external, etc. to call
>> code that is specifically NOT implemented in IDL? Why would you bother
>> writing a DLM if the functionality is already present in IDL?
>
> I'm thinking that he is doing part of his DLM processing in C, and then
> gets to a point where it would be Really Nice to call an IDL routine to
> finish it up because the IDL routine is at-hand and might be a pain to
> translate into C or whatever.
>
> It should be possible to refactor the code though to do some things in C
> and then return the intermediate results to IDL, where he can then call
> the IDL code. I can't speculate further without knowing more about the
> problem.

My IDL program implements a full graphical user interface using the IDL widget library, and is getting quite large (~ 80,000 lines of code). It must load a large number (~ 5000) of FITS files whose size sums up to 1 GB. We cannot alter these numbers. The first version of the program was completely written in IDL, and was really slow when loading files. So we are developing a DLM with C and the CFITSIO library, in order to make the loading process faster (our preliminary tests show a gain in time of more than one order of magnitude).

The program has a "log window" which keeps track of what the program is doing. The purpose of this window is to have a record of what it has been done. It is very important that every message is recorded in a file as well, and that the file is immediately flushed and closed after each message, in order for the messages to be saved even if the system breaks down during the loading process.

The log window is of course implemented using pure IDL. The problem is that the DLM should write a message in the log window for each FITS file that is loaded successfully. And in order to do this it should call a "write_log_message" procedure, which is obviously written in IDL (because it uses IDL widgets).

Thank you for your suggestions,
Maurizio.

Subject: Re: How to call an IDL procedure from a DLM
Posted by [Rick Towler](#) on Fri, 24 Feb 2006 16:27:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Maurizio Tomasi wrote:

> Karl Schultz wrote:
>> On Thu, 23 Feb 2006 08:26:54 -0500, Haje Korth wrote:
>>

<snip>

- > The log window is of course implemented using pure IDL. The problem is
- > that the DLM should write a message in the log window for each FITS file
- > that is loaded successfully. And in order to do this it should call a
- > "write_log_message" procedure, which is obviously written in IDL
- > (because it uses IDL widgets).

Could you not still refactor your code... You wouldn't even have to return the data to IDL while you are reading the files, just the status.

You could have a setup routine which you pass all of the filenames, a read_one routine which reads a single file and returns the status, and a data_return routine which returns all of the data to IDL. There would be other ways too. I think you could do this with little impact on your applications performance.

I believe you could also possibly use the external widget API (widget_stub and associated functions) to create an invisible widget that your DLM issues events to (IDL_WidgetIssueStubEvent()). Since the event struct only has a single long field that you define, you also need to write a routine that takes that value (say an address) and gets the status of your DLM function. All of this is in the EDG under internal API/Widgets.

-Rick
