Subject: Re: DLM's and C code
Posted by Craig Markwardt on Wed, 09 Jan 2002 06:14:17 GMT
View Forum Message <> Reply to Message

"Richard Tyc" <Richard_Tyc@sbrc.umanitoba.ca> writes:

> A somewhat IDL related question.
> I am trying to link in some C code via a DLM. I use a wrapper routine to
> handle the call from IDL and manipulate the args and return data. Within the
> wrapper, I call C functions linked in through another DLL.
>
> What is the best way to handle errors while deeply nested within layers of C
> functions.? The ANSI C code I am using essentially had exit(1) calls for
> major errors. Is there an IDL_ function (like say an exit handler) I can
> call to cleanly return to IDL rather than a trying to modify the call stack
> and get back to the IDL wrapper function to perform something like a return
> IDL_StrToSTRING("ERROR") ;

For example, do you mean the routine IDL_Message() ?

Looking in the EDG, there are lots of cool options you can use, which
define what happens after the error is triggered.

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Re: DLM's and C code
Posted by Richard Tyc on Wed, 09 Jan 2002 15:39:09 GMT
View Forum Message <> Reply to Message

I was thinking of IDL_Message too, but the descriptions of the 'actions'
possible puzzled me :
From Docs:

IDL_MSG_EXIT - Use this argument to cause the IDL process to exit after the
message is issued. This code should never be used in a system function or
procedure - it is intended for use in other sections of the system.

Is my C stub in the DLM/DLM a system function or procedure. Would this work
or would it crash IDL? I will try it

IDL_MSG_LONGJMP - It is an error to use this action code in code not called
by the IDL interpreter since the resulting call to longjmp() will be
invalid.
I assume this would be my case since I am deeply nested in the DLL and not
in the C function directly called by the interpreter.

So unless IDL_MSG_EXIT works, I don't think this will do. I will try it.

Thanks
Rich

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:onk7usoxty.fsf@cow.physics.wisc.edu...
>
> "Richard Tyc" <Richard_Tyc@sbrc.umanitoba.ca> writes:
>
>
>> A somewhat IDL related question.
>> I am trying to link in some C code via a DLM. I use a wrapper routine to
>> handle the call from IDL and manipulate the args and return data. Within
the
>> wrapper, I call C functions linked in through another DLL.
>>
>> What is the best way to handle errors while deeply nested within layers
of C
>> functions.? The ANSI C code I am using essentially had exit(1) calls
for
>> major errors. Is there an IDL_ function (like say an exit handler) I
can
>> call to cleanly return to IDL rather than a trying to modify the call
stack
>> and get back to the IDL wrapper function to perform something like a
return
>> IDL_StrToSTRING("ERROR") ;
>
> For example, do you mean the routine IDL_Message() ?
>
> Looking in the EDG, there are lots of cool options you can use, which
> define what happens after the error is triggered.
>
> Craig
>
> --
> -- --------------
> Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -- --------------

Subject: Re: DLM's and C code
Posted by Craig Markwardt on Wed, 09 Jan 2002 23:50:23 GMT
View Forum Message <> Reply to Message

"Richard Tyc" <richt@sbrc.umanitoba.ca> writes:

> I was thinking of IDL_Message too, but the descriptions of the 'actions'
> possible puzzled me :
> From Docs:
>
> IDL_MSG_EXIT - Use this argument to cause the IDL process to exit after the
> message is issued. This code should never be used in a system function or
> procedure - it is intended for use in other sections of the system.
>
> Is my C stub in the DLM/DLM a system function or procedure. Would this work
> or would it crash IDL? I will try it
>
> IDL_MSG_LONGJMP - It is an error to use this action code in code not called
> by the IDL interpreter since the resulting call to longjmp() will be
> invalid.
> I assume this would be my case since I am deeply nested in the DLL and not
> in the C function directly called by the interpreter.

Hi Rich--

You haven't really said what you want to do. In the first place you
said something about "cleanly returning to IDL." Well that has
several meanings.

I believe that

IDL_MSG_RET
Use this argument to make IDL_Message() return to the caller after
issuing the error message. In this case, the calling routine can
either continue or return to the interpreter as it sees fit.

is actually your best bet, because that forces a return to the calling
procedure in IDL. Then, presumably, you can have that procedure
handle the error appropriately. I think that IDL_MSG_EXIT is exactly
what you *don't* want, since there's nothing orderly about it at all.
It just shuts down IDL! Also, IDL_MSG_LONGJMP may work for you. My
interpretation of that message is that it returns to the command line
interpreter to await further commands.

Good luck,
Craig

--
-- --------------

Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: DLM's and C code
Posted by Richard Tyc on Fri, 11 Jan 2002 19:18:36 GMT
View Forum Message <> Reply to Message

Thanks Craig,

I am now using IDL_MSG_LONGJMP since in many cases IDL_MSG_RET does not get
me back to the C function called by the IDL interpreter (ie. the function
found in the *.dlm file which can be called from within IDL). Typically ,
the error is deeply nested in another c function inside a linked in DLL or
simply another C function not "visible" to IDL but in the same overall *.c
program compiled with MAKE_DLL.

I am a bit surprised LONGJMP works since the docs say :
"... It is an error to use this action code in code not called by the IDL
interpreter since the resulting call to longjmp() will be invalid. "

But, a seemingly safe return to the IDL command line occurs with the
message:
% Execution halted at: $MAIN$

Go figure....

Rich

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:onsn9fuls0.fsf@cow.physics.wisc.edu...
>
> "Richard Tyc" <richt@sbrc.umanitoba.ca> writes:
>
>> I was thinking of IDL_Message too, but the descriptions of the 'actions'
>> possible puzzled me :
>> From Docs:
>>
>> IDL_MSG_EXIT - Use this argument to cause the IDL process to exit after
the
>> message is issued. This code should never be used in a system function
or
>> procedure - it is intended for use in other sections of the system.
>>
>> Is my C stub in the DLM/DLM a system function or procedure. Would this
work

>> or would it crash IDL? I will try it
>>
>> IDL_MSG_LONGJMP - It is an error to use this action code in code not
called
>> by the IDL interpreter since the resulting call to longjmp() will be
>> invalid.
>> I assume this would be my case since I am deeply nested in the DLL and
not
>> in the C function directly called by the interpreter.
>
> Hi Rich--
>
> You haven't really said what you want to do. In the first place you
> said something about "cleanly returning to IDL." Well that has
> several meanings.
>
> I believe that
>
>   IDL_MSG_RET
>   Use this argument to make IDL_Message() return to the caller after
>   issuing the error message. In this case, the calling routine can
>   either continue or return to the interpreter as it sees fit.
>
> is actually your best bet, because that forces a return to the calling
> procedure in IDL. Then, presumably, you can have that procedure
> handle the error appropriately. I think that IDL_MSG_EXIT is exactly
> what you *don't* want, since there's nothing orderly about it at all.
> It just shuts down IDL! Also, IDL_MSG_LONGJMP may work for you. My
> interpretation of that message is that it returns to the command line
> interpreter to await further commands.
>
> Good luck,
> Craig
>
> --
>   -- --------------
> Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>   -- --------------

Subject: Re: DLM's and C code
Posted by Craig Markwardt on Sat, 12 Jan 2002 01:53:56 GMT
View Forum Message <> Reply to Message

"Richard Tyc" <Richard_Tyc@sbrc.umanitoba.ca> writes:
> Thanks Craig,
>

> I am now using IDL_MSG_LONGJMP since in many cases IDL_MSG_RET does not get
> me back to the C function called by the IDL interpreter (ie. the function
> found in the *.dlm file which can be called from within IDL). Typically ,
> the error is deeply nested in another c function inside a linked in DLL or
> simply another C function not "visible" to IDL but in the same overall *.c
> program compiled with MAKE_DLL.
>
> I am a bit surprised LONGJMP works since the docs say :
> "... It is an error to use this action code in code not called by the IDL
> interpreter since the resulting call to longjmp() will be invalid. "
>
> But, a seemingly safe return to the IDL command line occurs with the
> message:
> % Execution halted at: $MAIN$
>
> Go figure....

Hi Rich--

I'm still not sure you are getting it :-)

If you have a situation like this:
  IDL interpretter calls FUNC.PRO, which calls dlm function X, which
  calls Y which calls Z

and an error occurs in "Z" then here are the possibilities:
 * IDL_MSG_LONGJMP will always return to IDL, skipping X, Y and FUNC.PRO
   (by return to IDL I mean return to the IDL command line)
 * IDL_MSG_RET will always return to FUNC.PRO

If you want to return to X or Y, then skip all that IDL nonsense and
setup your own setjmp()/longjmp() calls.

Good luck,
Craig

--

 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------