## Subject: how to debug a IDL DLM routine Posted by xje4e on Tue, 12 Feb 2002 18:07:13 GMT

View Forum Message <> Reply to Message

Hi, big guys,

I have a question here:

how to debug a IDL DLM routine?

I know a method if you can make a IDL program as a .exe file. but how to make a IDL program as a .exe file?

In Visual C++ environment, in project->Settings->Debug, we can set executable for debug session. If we can make a IDL program as a executable file, we can debug it in VC step by step.

If anyone can give me a hint about how to make a IDL program as a .exe file. Or is there any other method to debug a IDL DLM routine with Visual C++?

Any suggestion will be appreciated.

Jujia Jin

Subject: Re: how to debug a IDL DLM routine
Posted by Gert Van de Wouwer on Wed, 13 Feb 2002 09:13:14 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message news:10ea38a6.0202121444.3c1d5b8a@posting.google.com...

- > Thanks, Richard,
- >
- > Here is my further question:
- >> I suggest going to your DLL project, and simply specifying the IDLDE
- >> executable.
- > How to specifying the IDLDE executable?

project->settings->debug: fill in idl\_path\idl32.exe

Subject: Re: how to debug a IDL DLM routine Posted by Richard Younger on Wed, 13 Feb 2002 14:37:29 GMT View Forum Message <> Reply to Message

Xiaoying Jin wrote:

> How to specifying the IDLDE executable?

Gert's post has the answer, namely project->settings->debug.

- > Do you mean you pause IDL for a while? But I think in this case all
- > you can debug and monitor is the variables in IDL not in C program. Am
- > I right? So how can I monitor C variable in this case?
- > How to attatch debugger to the IDLDE process?

Ah. As I said, this method is much dirtier. I wouldn't recommend it unless you have an error that you need to debug spontaneously for whatever reason. But it's the method I thought of when I first started debugging my DLMs. When I found out about the debug executable setting, I felt like giving myself a good whack upside the head, since I had been debugging the hard way for a couple months.

But if you want to try it, when you have the IDL prompt "press any key...", go to your VC++ environment, select Build->start debug->attach to process, and then select IDL.

You may have to select break, and then do a little twiddling to figure out where you need to go back to IDL and press a key to break out of the IDL\_GetKbrd() read loop. It'll be a tad confusing as there is no debugging info to IDL\_GetKbrd(), so you'll be looking at the assembly until you can pop up to your own code. Keeping the call stack visible helps with this. With a little playing, you can get back to your own code inside the VC++ debugger without too much difficulty.

Best of luck, Rich

--

Richard Younger

Subject: Re: how to debug a IDL DLM routine Posted by xje4e on Wed, 13 Feb 2002 21:25:01 GMT View Forum Message <> Reply to Message

Thank you so much for the help. In my understanding, I think specifying the IDLDE executable is still the best way as you suggested.

Best.

Julia Jin

Richard Younger <younger@II.mit.edu> wrote in message news:<3C6A7A29.F90C9E94@II.mit.edu>... > Xiaoying Jin wrote: > >> How to specifying the IDLDE executable? > Gert's post has the answer, namely project->settings->debug. > >> Do you mean you pause IDL for a while? But I think in this case all >> you can debug and monitor is the variables in IDL not in C program. Am >> I right? So how can I monitor C variable in this case? >> How to attatch debugger to the IDLDE process? > > Ah. As I said, this method is much dirtier. I wouldn't recommend it > unless you have an error that you need to debug spontaneously for > whatever reason. But it's the method I thought of when I first started > debugging my DLMs. When I found out about the debug executable setting. > I felt like giving myself a good whack upside the head, since I had been > debugging the hard way for a couple months. > But if you want to try it, when you have the IDL prompt "press any > key...", go to your VC++ environment, select Build->start debug->attach > to process, and then select IDL. > > You may have to select break, and then do a little twiddling to figure > out where you need to go back to IDL and press a key to break out of the > IDL\_GetKbrd() read loop. It'll be a tad confusing as there is no > debugging info to IDL\_GetKbrd(), so you'll be looking at the assembly > until you can pop up to your own code. Keeping the call stack visible > helps with this. With a little playing, you can get back to your own > code inside the VC++ debugger without too much difficulty. > > Best of luck, > Rich

Subject: Re: how to debug a IDL DLM routine Posted by xje4e on Wed, 13 Feb 2002 21:26:52 GMT View Forum Message <> Reply to Message

Thank you so much for the help. In my understanding, I think specifying the IDLDE executable is still the best way as you suggested.

Best,

Julia JIn

```
Richard Younger <younger@II.mit.edu> wrote in message
news:<3C6A7A29.F90C9E94@II.mit.edu>...
> Xiaoying Jin wrote:
>
>> How to specifying the IDLDE executable?
> Gert's post has the answer, namely project->settings->debug.
>
>> Do you mean you pause IDL for a while? But I think in this case all
>> you can debug and monitor is the variables in IDL not in C program. Am
>> I right? So how can I monitor C variable in this case?
>> How to attatch debugger to the IDLDE process?
>
> Ah. As I said, this method is much dirtier. I wouldn't recommend it
> unless you have an error that you need to debug spontaneously for
> whatever reason. But it's the method I thought of when I first started
> debugging my DLMs. When I found out about the debug executable setting,
> I felt like giving myself a good whack upside the head, since I had been
> debugging the hard way for a couple months.
>
> But if you want to try it, when you have the IDL prompt "press any
> key...", go to your VC++ environment, select Build->start debug->attach
> to process, and then select IDL.
>
> You may have to select break, and then do a little twiddling to figure
> out where you need to go back to IDL and press a key to break out of the
> IDL GetKbrd() read loop. It'll be a tad confusing as there is no
> debugging info to IDL GetKbrd(), so you'll be looking at the assembly
> until you can pop up to your own code. Keeping the call stack visible
> helps with this. With a little playing, you can get back to your own
> code inside the VC++ debugger without too much difficulty.
>
> Best of luck,
> Rich
```