
Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010

Posted by [ronn](#) on Wed, 25 Jan 2012 20:05:24 GMT

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

On Jan 25, 2:19 pm, b_gom <b_...@hotmail.com> wrote:

> I'm having trouble with the MS VC++ 2010 Express compiler when trying
> to debug a DLL called from an IDL DLM. I used to be able to use
> breakpoints in the C code and the debugger would stop on them as
> expected when running in debug mode.
>
> Lately, when I start the debugger in VC++, all the breakpoint markers
> turn to empty circles with the message: "The breakpoint will not
> current be hit. No symbols have been loaded for this document."
>
> I've searched and found dozens of threads on this topic in the general
> MS visual studio community, but no solutions.
>
> Anyone have any tips for getting debugging breakpoints to work within
> a DLM? I did have it working before in VC++ 2008 and 2010, but I don't
> know what's changed.
>
> Thanks

Which version of IDL are you running? And how do you start the debug process? Do you attach to an existing IDL process or spawn a new one?

Ronn Kling

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010

Posted by [b_gom](#) on Wed, 25 Jan 2012 22:33:42 GMT

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

On Jan 25, 1:05 pm, ronn <rlkl...@gmail.com> wrote:

> On Jan 25, 2:19 pm, b_gom <b_...@hotmail.com> wrote:
>
>
>
>
>
>> I'm having trouble with the MS VC++ 2010 Express compiler when trying
>> to debug a DLL called from an IDL DLM. I used to be able to use
>> breakpoints in the C code and the debugger would stop on them as
>> expected when running in debug mode.
>
>> Lately, when I start the debugger in VC++, all the breakpoint markers
>> turn to empty circles with the message: "The breakpoint will not

>> current be hit. No symbols have been loaded for this document."
>
>> I've searched and found dozens of threads on this topic in the general
>> MS visual studio community, but no solutions.
>
>> Anyone have any tips for getting debugging breakpoints to work within
>> a DLM? I did have it working before in VC++ 2008 and 2010, but I don't
>> know what's changed.
>
>> Thanks
>
> Which version of IDL are you running? And how do you start the debug
> process? Do you attach to an existing IDL process or spawn a new
> one?
>
> Ronn Kling
>
>

Hi Ronn,

this is IDL 8.1 on a windows 7 64-bit system, although I'm compiling a
32-bit DLM and calling it with 32-bit IDL. I have the debugger spawn a
new process using this command:

C:\Program Files\ITT\IDL\IDL81\bin\bin.x86\idlde.exe
with command arguments of:

-32

The debugger type is set to 'Auto'.

The only difference I can see from the second printing of your
'Calling C from IDL' book is that I have a post-build step that copies
the DLL to my default DLM path instead of putting it in the IDL bin
directory. The DLM loads and runs fine this way, its just that the
debugger doesn't seem to work.

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010
Posted by [Doug Edmundson](#) on Wed, 25 Jan 2012 23:00:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 1/25/12 3:33 PM, b_gom wrote:

> On Jan 25, 1:05 pm, ronnrkl...@gmail.com wrote:

>> On Jan 25, 2:19 pm, b_gom<b_...@hotmail.com> wrote:

>>

>>

>>

>>

>>

```

>>> I'm having trouble with the MS VC++ 2010 Express compiler when trying
>>> to debug a DLL called from an IDL DLM. I used to be able to use
>>> breakpoints in the C code and the debugger would stop on them as
>>> expected when running in debug mode.
>>
>>> Lately, when I start the debugger in VC++, all the breakpoint markers
>>> turn to empty circles with the message: "The breakpoint will not
>>> current be hit. No symbols have been loaded for this document."
>>
>>> I've searched and found dozens of threads on this topic in the general
>>> MS visual studio community, but no solutions.
>>
>>> Anyone have any tips for getting debugging breakpoints to work within
>>> a DLM? I did have it working before in VC++ 2008 and 2010, but I don't
>>> know what's changed.
>>
>>> Thanks
>>
>> Which version of IDL are you running? And how do you start the debug
>> process? Do you attach to an existing IDL process or spawn a new
>> one?
>>
>> Ronn Kling
>>
>>
>
> Hi Ronn,
>
> this is IDL 8.1 on a windows 7 64-bit system, although I'm compiling a
> 32-bit DLM and calling it with 32-bit IDL. I have the debugger spawn a
> new process using this command:
> C:\Program Files\ITT\IDL\IDL81\bin\bin.x86\idlde.exe
> with command arguments of:
> -32
> The debugger type is set to 'Auto'.
>
> The only difference I can see from the second printing of your
> 'Calling C from IDL' book is that I have a post-build step that copies
> the DLL to my default DLM path instead of putting it in the IDL bin
> directory. The DLM loads and runs fine this way, its just that the
> debugger doesn't seem to work.

```

When you run 64-bit IDL 8.1 workbench on Windows, IDL will run in-process. When you run 32-bit IDL 8.1 workbench on Windows, the workbench will run IDL in a separate executable called "idl_ipserver.exe". To debug, you'll need to attach to that process.

On Mac and Linux, 32 and 64-bit, the 8.x workbench runs in-process.

With IDL 7.x, the workbench always ran IDL out-of-process (idl_opserver.exe).

Hope this helps.

Cheers,
Doug
Exelis VIS

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010
Posted by [b_gom](#) on Wed, 25 Jan 2012 23:11:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Excellent, thanks for the timely response Doug.

The breakpoint marker still shows up as invalid when the debugger starts, but at least the breakpoint does catch when the code is called now.

I'm glad this is now working, but my next step is compiling a 64-bit DLM of the same routine. Do I understand correctly that for the 64-bit project I should set the debugger to launch IDL using C:\Program Files\ITT\IDL\IDL81\bin\bin.x86_64\idlde.exe ?

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010
Posted by [Doug Edmundson](#) on Wed, 25 Jan 2012 23:35:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 1/25/12 4:11 PM, b_gom wrote:

> Excellent, thanks for the timely response Doug.
>
> The breakpoint marker still shows up as invalid when the debugger
> starts, but at least the breakpoint does catch when the code is called
> now.
>
> I'm glad this is now working, but my next step is compiling a 64-bit
> DLM of the same routine. Do I understand correctly that for the 64-bit
> project I should set the debugger to launch IDL using C:\Program Files
> \ITT\IDL\IDL81\bin\bin.x86_64\idlde.exe ?
>

Whew, glad it worked. :)

I think the answer to your questions is "yes". Namely, if one wants to

debug 64-bit Windows IDL, one can launch "<IDL
DIR>\bin\bin.x86_64\idlde.exe" directly from Visual Studio. I say
"think" because I actually haven't tried it... I prefer to launch the
workbench normally and then attach moments later. However, the "launch
from Visual Studio" workflow is likely faster and I believe there are
some here at VIS who prefer it too. One might have to worry about how
their PATH and such is set up, but that would just be more project settings.

Just to explain the reason behind running "in" and "out" of process, we
initially wanted the workbench to run in-process because it solves
rendering issues, alleviates _some_ "on top" issues, etc. On all
platforms it works well, except 32-bit Windows. In that case we had to
move IDL out-of-process so that it would have more memory. The
workbench's JVM was fragmenting the process' memory too severely and the
largest chunk IDL could scrounge was only 400 MB or so. Hope this
rounds out the story a bit.

Cheers,
Doug
Exelis VIS

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010
Posted by [b_gom](#) on Thu, 26 Jan 2012 00:54:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

>
> Whew, glad it worked. :)

Well, mostly. Now I seem to get lingering idl_ipserver.exe processes
that I can't get rid of without rebooting. For example, if I hit a
breakpoint and then stop debugging, it seems that IDL doesn't get the
message to continue. If I then close IDL, then there is an
idl_ipserver process that I can't kill, even after closing IDL and
MSVC.

If I call the dll without debugging, everything cleans up properly.

What are the limitations on the debugger that I should be aware of
when starting, stopping, exiting?

Thanks

Subject: Re: Calling all DLM experts- debugging in MSVC++ 2010
Posted by [Doug Edmundson](#) on Thu, 26 Jan 2012 16:39:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 1/25/12 5:54 PM, b_gom wrote:

>>

>> Whew, glad it worked. :)

>

> Well, mostly. Now I seem to get lingering idl_ipserver.exe processes
> that I can't get rid of without rebooting. For example, if I hit a
> breakpoint and then stop debugging, it seems that IDL doesn't get the
> message to continue. If I then close IDL, then there is an
> idl_ipserver process that I can't kill, even after closing IDL and
> MSVC.

>

> If I call the dll without debugging, everything cleans up properly.

>

> What are the limitations on the debugger that I should be aware of
> when starting, stopping, exiting?

>

> Thanks

Hmmm. I do know that if MSVC has any hold on a process, you won't be able to kill the process. On Mac and Linux, the GDB debugger behaves the same. So, the only thing I can think of is to double check that MSVC is fully detached (Debug menu, "Detach All"). Failing that, double check that no MSVC process survives when MSVC is shut down.

The idl_ipserver.exe process does have a mechanism to self terminate when contact is lost with the workbench. That mechanism can take up to 20 seconds. I don't believe that could be part of the problem - at least we haven't seen it here with VS 2005 nor 2010.

Hope this helps,
Doug
Exelis VIS