
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
