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, 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.

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