

## RE: Memory increasing even when in debug-break

---

*Source:*

<http://www.tech-archive.net/Archive/VisualStudio/microsoft.public.vsnnet.debugging/2007-03/msg00051.html>

---

- *From:* [jetan@xxxxxxxxxxxxxxxxxxxxxx](mailto:jetan@xxxxxxxxxxxxxxxxxxxxxx) ("Jeffrey Tan[MSFT]")
  - *Date:* Fri, 16 Mar 2007 07:06:42 GMT
- 

Hi John ,

Based on my understanding, your complex pure managed C# application is experiencing memory leak which is indicated by "Private Bytes" counter from PerfMon. Even strange, your application event leaks while the debugger breaks your application. If I have misunderstood you, please feel free to tell me, thanks.

Actually, there are 2 types of memory leak in .Net: native leak and managed leak. The "Private Bytes" increases without reduce indicates that there is memory leak in your application. Then you should check if ".NET # of Bytes in all Heaps" counter also increases during the leak. If it also increases without reduce, it means a managed memory leak. Otherwise, there is a native leak(for example, in the FCL or even CLR native code). Please refer to the links below for more details:

"I have a memory leak!!! What do i do? (defining the "where")"

<http://blogs.msdn.com/tess/archive/2005/11/25/496899.aspx>

"Debugging Memory Problems"

<http://msdn2.microsoft.com/en-us/library/ms954591.aspx>

I recommend you to give it a check.

More interesting is the debugger breaks while memory still increasing issue. Can you tell me what debugger you are using to break your application, VS debugger or Windbg? If you are using VS debugger, the only possibility is caused by the activity of the debugger help thread. VS debugger leverages a special debugger help thread in debuggee process to get information of the managed process and to execute certain debugging operations. I suspect while the debugger breaks the debugger help thread is still allocating memory, which causes the memory increase. Anyway, this is merely a guess. Please refer to the link below for more details:

"Special threads in CLR "

<http://blogs.msdn.com/yunjin/archive/2005/07/05/435726.aspx>

"Implications of using a helper thread for debugging"

<http://blogs.msdn.com/jmstall/archive/2004/10/13/241828.aspx>

To prove my thought, I would recommend you to use windbg to attach and break your application, will the memory still increase? Windbg uses native

RE: Memory increasing even when in debug-break

Win32 debugging service. It normally issues a "int 3" software interrupt (software breakpoint) in the debuggee, which causes the CPU to stop all the threads (including the thread that performs GC, finalizer) in your application. Since there are no running threads in your process, the memory should not be touched by anybody (threads). You may give it a try.

Additionally, you may use Process Explorer to monitor if any threads are still activating while your debugger breaks the application. This will help to reveal which thread is responsible for the memory increasing problem.

I will wait for your further test result, thanks.

Best regards,  
Jeffrey Tan  
Microsoft Online Community Support

=====

Get notification to my posts through email? Please refer to <http://msdn.microsoft.com/subscriptions/managednewsgroups/default.aspx#notifications>.

Note: The MSDN Managed Newsgroup support offering is for non-urgent issues where an initial response from the community or a Microsoft Support Engineer within 1 business day is acceptable. Please note that each follow up response may take approximately 2 business days as the support professional working with you may need further investigation to reach the most efficient resolution. The offering is not appropriate for situations that require urgent, real-time or phone-based interactions or complex project analysis and dump analysis issues. Issues of this nature are best handled working with a dedicated Microsoft Support Engineer by contacting Microsoft Customer Support Services (CSS) at <http://msdn.microsoft.com/subscriptions/support/default.aspx>.

=====

This posting is provided "AS IS" with no warranties, and confers no rights.

.