

Re: What does GC.GetTotalMemory really tell us?

Source: <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2004-04/1338.html>

From: Jon Skeet [C# MVP] (skeet_at_pobox.com)

Date: 04/20/04

Date: Tue, 20 Apr 2004 15:06:23 +0100

Daniel Billingsley <dbillingsley@NO.durcon.SPAAMM.com> wrote:

> *I did try using GC.Collect() and it didn't make any difference. I didn't*
> *add the second step.*
>
> *I also had noted that the documentation is conspicuously incomplete in*
> *describing the exact effect of the true parameter.*
>
> *I'm still curious why the memory is apparently never deallocated even when*
> *the variable is set to null. Or, in the case of the actual application*
> *where I was using it, when the form's Close() is called. I mean in that*
> *case even a minute later. I thought the whole point of being a good little*
> *Disposing boy was to free up memory resources, but if GetTotalMemory is*
> *telling us anything at all it's not nearly so simple.*
>
> *Would the following be correct?*
>
> *Calling Dispose() does not guarantee that a resource will be freed, it just*
> *guarantees that the GC will be able to free it if necessary. The GC may*
> *very well only free it if in fact it does believe it necessary. Maybe it*
> *makes sense that the memory would remain allocated as long as it wasn't*
> *needed as it would help performance if the item was needed again later.*

Dispose doesn't really say anything about GC – it isn't usually used for managed resources. It's more usually used for unmanaged resources or references to unmanaged resources.

I suspect the more likely culprit for the "strange" results is that you're actually allowing the garbage collector to collect before you think anyway. For instance:

```
MemoryHog hog = new MemoryHog();
howMuchDuring = GC.GetTotalMemory(true);
Console.WriteLine("Memory with hog:\t{0}", howMuchDuring);
hog = null;
```

There's nothing to stop the GC from collecting the new object in the second line – the local variable isn't read at any time after its initial assignment.

microsoft.public.dotnet.framework: Re: What does GC.GetTotalMemory really tell us?

--

Jon Skeet - <skeet@pobox.com>

<http://www.pobox.com/~skeet>

If replying to the group, please do not mail me too