

Re: GC and dispose

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-06/msg00599.html>

- *From:* "Daniel Billingsley" <DanielBillingsley@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 21 Jun 2006 08:46:22 -0400
-

"Petros Amiridis" <petros@xxxxxxxxxxxx> wrote in message
[news:OyV\\$mvOIGHA.1204@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:OyV$mvOIGHA.1204@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Phil Wilson wrote:

You example doesn't help because there's nothing in Foo that needs disposing, and Dispose isn't related to GC as you seem to be assuming.

What if it does have something that needs disposing? An if Dispose doesn't have anything to do with GC, could tell me briefly what are good practices that can make your application consume less memory?

I've noticed that my application starts and consumes several times more memory than equivalent native applications. The problem is that whenever I open/close windows in my application, the memory that it occupies grows more. I wait and nothing happens. Shouldn't GC release the memory as long as an object is not referenced any more?

I think to some degree it's a waste of time trying to help GC do its job better in the way it seems you're thinking, as it is a fairly indeterminate mechanism in the first place. It is somewhat important in some cases to help GC have less to do though, because it can become a performance issue. For example, using lots of stringbuilders for what are really just simple concatenations. But that really manifests itself as a performance issue more than a memory one in my testing.

As somebody else said, the Dispose pattern is for making sure UNMANAGED resources inside your classes get cleaned up. That is, making sure there are no memory leaks in things that GC doesn't try to deal with. That's why the statement "Dispose doesn't have anything to do with GC".

I have no idea what "several times more memory than equivalent native applications" means. Equivalent how –Functionality? Code? That's such a

Re: GC and dispose

nebulous comparison I don't think I'd spend much time worrying about that either.

Is there a specific problem you're having or specific reason you think you may have a memory leak in the application?

.