

Re: Garbage collector bug

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.clr/2004-03/0155.html>

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

Date: 03/21/04

Date: Sun, 21 Mar 2004 17:51:50 -0000

Shital Shah <shital@orioninc.com> wrote:

> *I think this chain of thoughts is missing a point. Continuing with your
> analogy of subscribing a mailing list, in perfectly working world, yes I
> need to inform list server that I'm dead or I have moved so it can remove me
> from it's list. But the entire purpose of having a .Net garbage collector is
> that I never have to worry about memory deallocation if I'm in 100% managed
> world. So I'm supposed to allocate memory as and when I need [equ to
> subscribing a list] but can get away with having it deallocated myself [equ
> to unsubscribing]. In terms of your analogy, if mailing list finds out that
> mails are being returned because I'm dead or have moved, it should
> automatically unsubscribe me if it works .Net way. If it works C++ way, yes, I
> need to tell that server. In fact this is the way most mailing list such as
> Yahoo! Groups works today.*

But the "mails" aren't being returned. The event can still be fired and the delegate executed perfectly well.

> *Having said that, the important point is that once I set variable to null,
> I've lost the reference. There is NO way I can get that reference back and
> so keeping with GC theory, it MUST be collected because it's not accessible
> by anyway.*

No, it **is** accessible – by the delegate. There's nothing wrong with that.

> *The CLR delegate implementation is simply consuming this
> reference for it's internal working which programmer should not care about
> neither should expect to work around by forcing IDisposable on everything in
> their code that listens to some events.*

It's nothing to do with the delegate implementation – the interface itself exposes the target of the delegate.

> *Just a thought that there is a dead
> object seating in my memory listening to events and doing funny things
> without my knowledge is scary and can produce hard to debug behavior.*

microsoft.public.dotnet.framework.clr: Re: Garbage collector bug

Who's to say it's a dead object? Just because you have no more references to it yourself doesn't mean it's necessarily dead.

> *I certainly can explain why it's happening but there is no excuse that*
> *it should not be.*

Well, it's your opinion that it shouldn't work that way – but personally I'm very glad that it does.

--

Jon Skeet - <skeet@pobox.com>

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

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