

Re: Performance on string destruction

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-08/msg04250.html>

- *From:* Bob Grommes <bob@xxxxxxxxxxxxxxxx>
 - *Date:* Mon, 22 Aug 2005 15:02:51 -0700
-

Setting it to null will accomplish nothing unless perhaps there is a lot of other processing in the same method before it exits. Otherwise, just let the StringBuilder reference go out of scope when the routine exits.

Deleting what's in the buffer would only make sense if you're going to reuse it. Setting the StringBuilder instance to null will cause the GC to reclaim the whole object, including the buffer, *at such time as the GC feels it's necessary*.

I emphasized the last point because so many people get bogged down in anxiously watching memory (often looking at the wrong memory indicator, as well) and then have a cow when it doesn't (appear to) be freed up. In effect, memory is marked by the GC as reclaimable when it's no longer needed.

--Bob

SevDer wrote:

Thinking for performance & system resources;

I have a string field inside a major object that contains big XML response.
When I am done with it, I want to clear out this field to give some relief to memory.

Should I use this:

```
SabreContractResponse.Remove(0, SabreContractResponse.Length);  
SabreContractResponse = null;
```

or just this:

```
SabreContractResponse = null;
```

Re: Performance on string destruction

or do you have any other suggestion?

Thanks