

Re: delete vs delete[]

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2007-05/msg00533.html>

- *From:* "Ben Voigt" <rbv@xxxxxxxxxxxxxx>
 - *Date:* Tue, 22 May 2007 08:28:55 -0500
-

"Carl Daniel [VC++ MVP]" <cpdaniel_remove_this_and_nospam@xxxxxxxxxxxxxx>
wrote in message news:O5a9A9CnHHA.404@xxxxxxxxxxxxxx

Alex Blekhman wrote:

Carl Daniel [VC++ MVP] wrote:

bob@xxxxxxxxxxxxxx wrote:

I was just wondering why the compiler won't warn you when you try to do this:

```
TCHAR *t = new TCHAR[20];  
delete t;
```

A fair question – I can't see any reason why the compiler couldn't warn in a case like this. Of course, in real programs, there's likely to be a much greater distance between the allocation and the deallocation, which would make generating an error or warning considerably harder.

I think, it is because the task of a compiler is to check syntactical correctness of a code, while it's up to developer to check logical correctness. However, Team System edition of VC++ has handy `/analyze` compiler option, which besides other checks, attempts to detect new/delete mismatch. See, for example, warnings C6278, C6283.

I guess my wording was too clever for my own good :) There's lots of reasons why a compiler `_wouldn't_` warn on something like this, but no reason why it `_couldn't_`.

Re: delete vs delete[]

It is much more appropriate for tools like lint or team system's /analyze to detect stuff like this, definitely!

Why? The only reason I can see (not an invalid one to be sure) is so MS and others can make money charging extra for the advanced analysis. Certainly most warnings now emitted by the compiler were once the responsibility of lint. I don't think anyone would now argue having them in lint is "much more appropriate".

-cd