

## Malloc, free and thread-safety

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2004-09/0269.html>

---

**From:** RobinJMoore (*RobinJMoore\_at\_discussions.microsoft.com*)

**Date:** 09/09/04

Date: Thu, 9 Sep 2004 10:35:07 -0700

I've modified a large and complex Visual C program to make it multithreaded.  
I get intermittent exceptions such as:

```
"Heap corruption detected at 0316D980  
HEAP[recv.exe]: HEAP: Free Heap block 316d978 modified at 316d988 after it  
was freed"  
"First-chance exception in recv.exe (NTDLL.DLL): 0xC0000005: Access  
Violation."  
+ various other Access Violations
```

Within each thread, I've checked carefully that malloc() and free() are  
being used correctly, i.e. for every malloc() there is a corresponding free()  
and they are called in the right order.

The exceptions are infrequent and intermittent for basically the same traces  
of execution. Sometimes the program runs for long periods and completes  
successfully. I suspect that there is some problem relating to malloc() and  
free() not being thread-safe. So my questions are:

- 1) Are malloc(), calloc() and free() thread safe?
- 2) Are there any special measures I should be employing to use malloc() and free() in a multithreaded program?
- 3) Depending on the answers to 1) and 2), should I be restricting access to malloc() and free() using some synchronisation method such as semaphores?
- 4) Is there a good way of debugging heap related problems for multithreaded programs?

Unfortunately there is too much code to post a specific example.

Thanks.