
Subject: Re: CTRL + C = 659 Heap leaks
Posted by [mirek](#) on Tue, 16 Jul 2019 07:38:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 16 July 2019 02:11 mirek wrote on Mon, 15 July 2019 18:11
Yeah, but then I will not catch leaks in global objects.

I do not understand why. Just do not show memory leak report when an object on stack wasn't destroyed.

Maybe we are nor speaking about the same thing...

```
struct Foo {  
    byte *ptr;  
  
    ~Foo() { if(ptr) delete[] ptr; } // this might be missing  
};  
  
Foo foo;  
  
CONSOLE_APP_MAIN {  
    foo.ptr = new byte[1000];  
}
```

If you couple leaks detection to stack, either that leak is wrongly ignored or not leak wrongly reported (depending on what exactly you do).

Really the only sensible thing to do is either somehow make Ctrl+C call Exit by installing exception handler in Win32, but that is hard as the handler is running in different thread, or make it disable leak checks completely in Win32 in Ctrl+C exception handler.
