

## Re: Can I Increase size of nNoMansLandSize?

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2008-01/msg01099.html>

---

- *From:* Joseph M. Newcomer <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)>
  - *Date:* Wed, 23 Jan 2008 00:02:33 -0500
- 

One thing I do is put into my CWinApp-derived class an OnIdle handler

```
long CMyApp::OnIdle(int count)
{
    ASSERT(_heapchk() == _HEAPOK);
    return CWinApp::OnIdle(count);
}
```

That's my first and best step for looking for heap damage; it isolates it to which event is responsible. Next, I start adding that ASSERT in other places in the sequence between where it was OK and where it failed.

joe

On Wed, 16 Jan 2008 11:23:48 -0700, "Chris Shearer Cooper" <[chris\\_web@xxxxxxx](mailto:chris_web@xxxxxxx)> wrote:

It's not writing to address 0x00000001, it's writing the DWORD value 0x00000001 to a random address.

The way I know, is that I will get an MFC error "heap corruption" and when I look at memory, I see that in the "no man's land" that the debug memory allocation creates, one of the DWORDs has been changed to 0x00000001.

Or, sometimes, I get a weird error because a pointer is all screwy, and when I look, the pointer has been changed from a reasonable value to 0x00000001.

"AliR (VC++ MVP)" <[AliR@xxxxxxxxxxxxx](mailto:AliR@xxxxxxxxxxxxx)> wrote in message [news:3iqjj.33373\\$4V6.17121@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:3iqjj.33373$4V6.17121@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

If your program tries to write to address 0x00000001 VStudio will stop the program at the line when that happens, if you are running it in the debugger.

How do you know that it is writing to 0x00000001?

AliR.

"Chris Shearer Cooper" <[chris\\_web@xxxxxxx](mailto:chris_web@xxxxxxx)> wrote in message

Re: Can I Increase size of nNoMansLandSize?

news:13osb6htofdcg4f@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I run in debug mode nearly all the time, so that's when I see it, but I'm thinking that if this happens in release mode too, we wouldn't know ... it would show up as some random crash for an unknown reason, or an unexpected data corruption, something like that ...

"AliR (VC++ MVP)" <AliR@xxxxxxxxxxxx> wrote in message  
news:URpjj.33366\$4V6.9143@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Does this only happen in Release mode?

AliR.

"Chris Shearer Cooper"  
<chris\_web@xxxxxxx> wrote in message  
news:13oqkn42often9a@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

My application, for some reason, is writing a single DWORD into a random location in its memory (the value is always 0x00000001), and I'm trying to find out where!

I can't find any rhyme or reason to when this happens, so right now the best I can do is stick a bunch of calls to \_CrtCheckMemory() all over the program and hope that the corruption will (1) hit a memory block in a way that \_CrtCheckMemory() can see it, and (2) that I will eventually be able to narrow down \_when\_ this is happening.

But I had a thought. If I

Re: Can I Increase size of nNoMansLandSize?

could tap into the MFC  
debug allocation  
scheme, and increase the  
size of nNoMansLandSize,  
then the chance of  
the bug hitting an area of  
memory that I could detect  
would increase  
greatly!

But there doesn't seem to be  
any way to do this, outside  
of rebuilding  
the MFC libraries, which I  
guess is a possibility, but  
not one I'm  
looking forward to.

I can't find any other way to  
tap into the new/debug  
scheme. I tried  
overriding new and delete,  
but got a bunch of "symbol  
already defined"  
errors. I was thinking if I  
could tap into that, I could  
just increase  
the size of every allocation  
(so if the caller wanted 4  
bytes, I'd  
actually allocate 40 and fill  
the extra 36 in with  
something I could  
check later).

Any suggestions as to the  
least painful way to track  
down this problem?

Thanks!  
Chris

Re: Can I Increase size of nNoMansLandSize?

Joseph M. Newcomer [MVP]

email: [newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)

Web: <http://www.flounder.com>

MVP Tips: [http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)

.