

Re: Leak after using setlocale

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.stl/2005-02/0058.html>

From: Doug Harrison [MVP] (dsh_at_mvps.org)

Date: 02/11/05

Date: Fri, 11 Feb 2005 11:59:31 -0600

Peter wrote:

>Sorry I didn't say what I used.

>

>I used Task Manager, because the loop is tight, you can watch it
>escalate in the process tab of task manager (if you have the Memory

>Usage, Memory Delta and VM columns selected).

>

>If you comment out the call to setlocale the process has a steady

>memory usage.

After increasing your loop limit from 100 to 1000000, I see what you mean. It does look like a leak. I tried using the _CrtXXX heap debugging functions but was unable to find a leak. Putting your fstream stuff in its own block and dumping leaks around it reveals no leaks at all; setting a checkpoint before the loop and dumping leaks after the fstream block yields endless identical output that looks like this:

Dumping objects ->

{97} normal block at 0x00324500, 512 bytes long.

Data: < > 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00

{74} normal block at 0x00324A90, 16 bytes long.

Data: <@wL > 40 77 4C 10 01 00 00 00 00 00 00 00 00 00 00 00

{60} normal block at 0x00322E10, 33 bytes long.

Data: < C -----> 00 43 00 CD CD CD CD CD CD CD CD CD CD CD CD CD

{59} normal block at 0x00322DB8, 40 bytes long.

Data: < |L > 14 7C 4C 10 16 00 00 00 00 00 00 00 00 00 00 00

Object dump complete.

So the debug heap is too blunt an instrument to determine what's going on. Possibly the library is using special allocation techniques that bypass debug heap tracking, or it's acquiring resources directly from Windows that it isn't releasing. Perhaps something like Purify could help pinpoint the bug.

As a workaround, instead of specifying LC_CTYPE, you can use LC_ALL. In addition, I did not observe this problem in VC 7.1.

microsoft.public.vc.stl: Re: Leak after using setlocale

--

Doug Harrison
Microsoft MVP - Visual C++