# How can I force memory to be allocated above the 4GB boundary (for debugging purposes) on Windows 7?

devblogs.microsoft.com/oldnewthing/20161014-00

October 14, 2016

Raymond Chen

A customer discovered that their program has 64-bit-to-32-bit pointer truncation bugs. They are working hard to track down all the places where this happens. On Windows 8 and higher, they can build their program with high entropy VA to make ASLR be super-aggressive about putting heap and stack memory above the 4GB boundary. However, they still have some customers running Windows 7, and on Windows 7, even though ASLR randomizes the location of the heap and stack, it does not push them above the 4GB mark.

The customer asked for suggestions on what steps they can take to help flush out their pointer truncation bugs on Windows 7 machines.

Unfortunately, there's no easy solution for Windows 7. You can set the system `Allocation-Preference` to `MEM_TOP_DOWN`. This requires a reboot, and the setting applies to the entire system (not a single process), and it makes memory allocation slower, so this approach is suitable only for testing purposes.

The customer wanted to force the heap and stack above 4GB on production machines, so the above approach is not recommended. We saw in the earlier discussion that there exists a poor-man's solution: Reserve all the available memory below the 4GB mark. That will push new allocations above the 4GB mark. The initial stack and initial heap segment will still be below 4GB, and future allocations in the process will be slower because it increases the amount of address space under management. (Searching for an open address takes longer.) The exact impact will vary depending on the application's memory allocation patterns, so you will want to measure the performance before and after to see if the degradation is acceptable. But it's better than nothing,

Raymond Chen

**Follow**