

# Adventures in application compatibility: The bogus memory calculation

[devblogs.microsoft.com/oldnewthing/20160822-00](https://devblogs.microsoft.com/oldnewthing/20160822-00)

August 22, 2016



Raymond Chen

One of my colleagues shared with me one of his application compatibility stories.

There was a program that would fail on some computers but not others, and it wasn't clear why. The problem was traced to the size of an internal cache. Now, `GlobalMemoryStatus` officially returns unsigned values, but if the calling application is not marked /LARGE-ADDRESSAWARE, then `GlobalMemoryStatus` reports a maximum of `2GB - 1` bytes of memory, for compatibility purposes.

You'd think that this would be enough to keep old programs happy, but apparently not. This particular program wasn't content with the values that it got from `GlobalMemoryStatus`. Instead, it took the `dwTotalPhys` and added it to the `dwTotalPageFile`, and treated the result as a signed value. This means that on systems with more than 2GB of memory, the addition will produce a total of `0xFFFFFFFF`, which is a negative value when interpreted as a signed result, which in turn causes the program to crash.

My colleague fixed the program by patching out the instructions that added `dwTotalPhys` to `dwTotalPageFile`, and had the program operate solely on `dwTotalPhys`, which is probably what it should have been operating on in the first place.

You see, even though the field in the MEMORYSTATUS structure is named `dwTotalPageFile`, it doesn't actually give you the size of the page file. The documentation of `dwTotalPageFile` says

The current size of the committed memory limit, in bytes. This is physical memory plus the size of the page file, minus a small overhead.

Yes, this is a case of bad naming. (You can come up with your own theories how we ended up with the bad name.)

By adding `dwTotalPhys` and `dwTotalPageFile`, the code was double-counting the physical memory.

The conclusion my colleague drew from this exercise was that there are still programmers out there who are working hard to skip the documentation, come up with bad ideas, and implement them poorly.

I admire the program's dedication to getting everything wrong despite the operating system's efforts to save them from themselves.

Raymond Chen

**Follow**

