

How do I prefetch data into my memory-mapped file?

 devblogs.microsoft.com/oldnewthing/20120601-00

June 1, 2012



Raymond Chen

A customer created a memory mapping on a large file and found that when the memory manager wanted to page in data from that file, it did so in **32KB** chunks. The customer wanted to know if there was a way to increase the chunk size for efficiency. The memory manager decides the chunk size for memory-mapped files, and the chunk size is currently set to eight pages, which on a system with 4KB pages, comes out to 32KB chunks. (Note that this chunk size is an internal parameter and is subject to change in future versions of Windows. I'm telling a story, not providing formal documentation.) You have a few options. The first option is to switch from memory-mapped files to explicit disk I/O. If you do that, then you have full control over the chunk size. It also means that you have finer control over I/O errors, because you will be told of the error in a controlled manner. As opposed to waiting for the exception to occur, and then carefully parsing the exception to verify that it was in your memory-mapped file region (and not in some other part of the address space), and then trying to unwind out of the exception without crossing any frames that are outside your control. Many people decide not to go this route and stick with the memory-mapped file approach, not because they are really good at writing exception handlers and unwinding safely, but because they really like the convenience of memory-mapped I/O, and if something goes wrong with the I/O, they're fine with the program simply crashing. (Of course, there's the group of people who try to write the really clever exception handler and end up making a bigger mess when they mess up.)

Another option is to go ahead and create your memory-mapped file, but when you are about to do that thing that you want to trigger large-chunk I/O, you can issue sequential **Read-File** calls from the same file handle into a dummy buffer of, say, 1 megabyte. Do this before you start accessing the memory-mapped version of the file. This will "prefetch" the data off the disk into memory in the large chunks you desire (at a cost of some extra memcopy's).

[Raymond Chen](#)

Follow

