## Fragmentation

Heap fragmentation occurs when objects are collected from the Large Object Heap and new objects are allocated in the leftover space. If there are pinned objects in the heap, compacting is not performed and gaps are left behind. More to the point, new objects will not fit into the gaps (and are preferentially allocated to the end of the heap anyway), and so the allocated memory expands. In long-running applications with large objects, this can result in `OutOfMemory` exceptions despite the availability of unused memory, because there just isn't a large enough block of memory for an object to fit in. This is typically a Generation 2 / LOH problem (as we discussed in Chapter 2, *The Simple Heap Model)*, and it can be detected by comparing free blocks of memory versus the total size of memory.

Some fragmentation will always occur in the life of an application. Avoiding pinning will help avoid this problem. If you're not aware, pinning occurs when using fixed variables or fixed-size buffers. This creates a guaranteed location and size for a variable, which can be passed to an external, unmanaged DLL.