Understanding memory problems

Before you can fix a memory problem, you need to identify which kind of memory problem you have.These are the problems you're most likely to encounter, with a list of symptoms to help you recognise them:

Problem: Large object heap fragmentation

Symptoms:

- There is a lot of memory allocated to your application which isn't being used (ie hasn't been committed).
- *OutOfMemory* exceptions are thrown, although there's still spare space on the .NET heap. The exception is thrown unpredictably.
- The Memory fragmentation section on the summary page suggests you may have a problem with fragmentation.

Problem: Unmanaged memory leaks

Symptoms:

- Performance degrades while the application runs. Performance recovers when the application restarts, and then degrades again.
- On the timeline, the number of private bytes increases more quickly than the number of bytes in the .NET heap.
- Something other than the CLR is using a lot of unmanaged memory. Unmanaged memory usage grows or stays constant, even when your application has finished using unmanaged modules.

Problem: Managed memory leaks

Symptoms:

- Performance degrades while the application runs. Performance recovers when the application restarts, and then degrades again.
- The number of bytes in the .NET heap increases continually over time.
- Memory use increases whenever you carry out a specific action. Memory use doesn't go back to the same level after the action is complete.

Problem: Excessive overall memory usage

Symptoms:

- Every application has different memory usage and requirements, so there's no standard amount that is "too much" memory.
- Users may notice that the application is using more memory than they expect, especially in memory-critical environments.

If you aren't familiar with memory profiling, you might find our .NET Memory Primer useful before you start.