CHAPTER 4

Garbage Collection

In this chapter, we will examine the .NET garbage collector (GC), one of the primary mechanisms af f ecting the perf ormance of .NET applications. While f reeing dev elopers f rom worry ing about memory deallocation, the GC introduces new challenges f or constructing deterministically well-behav ing programs in which perf ormance is paramount. First, we will rev iew the ty pes of GC av ailable in the CLR, and see how adapting an application to the GC can be v ery benef icial in terms of ov erall GC perf ormance and pause times. Next, we'll see how generations af f ect GC perf ormance and how to tune applications accordingly . Toward the end of the chapter we will examine the APIs av ailable f or controlling the GC directly , as well as the subtleties inv olv ed in correctly using non-deterministic f inalization.

Many of the examples in this chapter are based on the authors' personal experience with real-world sy stems. Where possible, we tried to point y ou to case studies and ev en sample applications that y ou can work on while reading this chapter, illustrating the primary perf ormance pain points. The "Best Practices" section toward the end of the chapter is f ull of such case studies and examples. Howev er, y ou should be aware that some of these points are dif f icult to demonstrate with short code snippets or ev en a sample program because the situations where perf ormance dif f erences arise are ty pically within large projects that hav e thousands of ty pes and millions of objects in memory .