

Username: Pralay Patoria **Book:** Pro .NET Performance. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Introduction

This book has come to be because we felt there was no authoritative text that covered all three areas relevant to .NET application performance:

- Identifying performance metrics and then measuring application performance to verify whether it meets or exceeds these metrics.
- Improving application performance in terms of memory management, networking, I/O, concurrency, and other areas.
- Understanding CLR and .NET internals in sufficient detail to design high-performance applications and fix performance issues as they arise.

We believe that .NET developers cannot achieve systematically high-performance software solutions without thoroughly understanding all three areas. For example, .NET memory management (facilitated by the CLR garbage collector) is an extremely complex field and the cause of significant performance problems, including memory leaks and long GC pause times. Without understanding how the CLR garbage collector operates, high-performance memory management in .NET is left to nothing but chance. Similarly, choosing the proper collection class from what the .NET Framework has to offer, or deciding to implement your own, requires comprehensive familiarity with CPU caches, runtime complexity, and synchronization issues.

This book's 11 chapters are designed to be read in succession, but you can jump back and forth between topics and fill in the blanks when necessary. The chapters are organized into the following logical parts:

- Chapter 1 and Chapter 2 deal with performance metrics and performance measurement. They introduce the tools available to you to measure application performance.
- Chapter 3 and Chapter 4 dive deep into CLR internals. They focus on type internals and the implementation of CLR garbage collection—two crucial topics for improving application performance where memory management is concerned.
- Chapter 5, Chapter 6, Chapter 7, Chapter 8, and Chapter 11 discuss specific areas of the .NET Framework and the CLR that offer performance optimization opportunities—using collections correctly, parallelizing sequential code, optimizing I/O and networking operations, using interoperability solutions efficiently, and improving the performance of Web applications.
- Chapter 9 is a brief foray into complexity theory and algorithms. It was written to give you a taste of what algorithm optimization is about.
- Chapter 10 is the dumping ground for miscellaneous topics that didn't fit elsewhere in the book, including startup time optimization, exceptions, and .NET Reflection.

Some of these topics have prerequisites that will help you understand them better. Throughout the course of the book we assume substantial experience with the C# programming language and the .NET Framework, as well as familiarity with fundamental concepts, including:

- Windows: threads, synchronization, virtual memory
- Common Language Runtime (CLR): Just-In-Time (JIT) compiler, Microsoft Intermediate Language (MSIL), garbage collector
- Computer organization: main memory, cache, disk, graphics card, network interface There are quite a few sample programs, excerpts, and benchmarks throughout the book. In the interest of not making this book any longer, we often included only a brief part—but you can find the whole program in the companion source code on the book's website.

In some chapters we use code in x86 assembly language to illustrate how CLR mechanisms operate or to explain more thoroughly a specific performance optimization. Although these parts are not crucial to the book's takeaways, we recommend dedicated readers to invest some time in learning the fundamentals of x86 assembly language. Randall Hyde's freely available book "The Art of Assembly Language Programming" (<http://www.artofasm.com/Windows/index.html>) is an excellent resource.

In conclusion, this book is full of performance measurement tools, small tips and tricks for improving minor areas of application performance, theoretical foundations for many CLR mechanisms, practical code examples, and several case studies from the authors' experience. For almost ten years we have been optimizing applications for our clients and designing high-performance systems from scratch. During these years we trained hundreds of developers to think about performance at every stage of the software development lifecycle and to actively seek opportunities for improving application performance. After reading this book, you will join the ranks of high-performance .NET application developers and performance investigators optimizing existing applications.

Sasha Goldshtein

Dima Zurbalev

Ido Flatow