C# Multithreaded And Parallel Programming

Develop powerful C# applications to take advantage of today's multicore hardware

Rodney Ringler

Download EBOOK
Synopsis

Develop powerful C# applications to take advantage of today’s multicore hardware

About This Book
Make use of the latest Visual Studio debugging tools, to manage and debug multiple threads running simultaneously
Learn how to use the Thread, Task, and Parallel libraries in your C# applications
Explore the evolution of multithreaded development in C#, starting with BackgroundWorker classes and moving on to threads and tasks and finally covering Async

Who This Book Is For
If you are a C# developer and want to learn how to take advantage of the features of .NET for concurrent and multithreaded applications, then this book is for you. If you are already comfortable with C# but want to learn more about parallel design patterns, threads, tasks, and async, then look no further!

What You Will Learn
Explore all the essential methods used for programming multithreaded applications
Enhance the performance of an application by designing various parallel operations to achieve concurrency
Build powerful applications using the Task Parallel Library (TPL), which makes concurrent processing of items in a data collection simple
Implement data parallelism using the Parallel library, concurrent collections, and PLINQ
Debug your multithreaded applications using the Threads view, Tasks window, Parallel Stacks window, and Parallel Watch window
Accomplish any given parallel task using two of the most popular parallel patterns for development: Pipelining and producer-consumer
Get to grips with the Asynchronous Programming Model (APM) to learn to begin and end asynchronous operations

In Detail
Most modern machines have dual-core processors. This means that the present-day computer has the ability to multitask. Using multiple cores means your applications can process data faster and be more responsive to users. However, to fully exploit this in your applications, you need to write multithreading code.

We will begin by covering some techniques that have been around since the beginning of .NET, including the BackgroundWorker component, timers, and the Thread class. We will use tasks, task factories, and parallel loops to develop multithreaded applications at a higher level than directly creating and managing individual threads. Finally, we will look at the tools Visual Studio provides for debugging parallel applications, common concurrent design patterns, and the latest updates in PLINQ and async.

Book Information
Paperback: 482 pages
Publisher: Packt Publishing - ebooks Account (December 24, 2014)
Language: English
ISBN-10: 184968832X
As C# continues to evolve the need for appropriate multi-threading capability expands to meet the needs of the programmer. This book collates the requirements and history of multi-threaded programming in C# and introduces the advancements beyond use of the BackgroundWorker thread programming that have been introduced since 2010. The advancements include light, heavy, advanced, and TaskParallel thread programming. I have been using threads in my application programming since 1995 and this is the most complete and well written book on managing them properly that I have encountered in all that time, thus the five stars. Every enterprise software design job interview includes a heavy set of questions on thread programming no matter what language is being used. Even Java 8 is using similar object naming for threading procedures. So, reading this book will give you an expert understanding of modern thread programming. If you are serious about being an expert on thread programming, buy this book.

There are some errors, that needs to be corrected. If you can correct the errors, it will be a good book.

Download to continue reading...
