Parallel Programming with Microsoft®.NET
Design Patterns for Decomposition and Coordination on Multicore Architectures

Colin Campbell
Ralph Johnson
Ade Miller
Stephen Toub
Contents

Foreword xi
Tony Hey

Preface xiii
Who This Book Is For xiii
Why This Book Is Pertinent Now xiv
What You Need to Use the Code xiv
How to Use This Book xv
   Introduction xvi
   Parallelism with Control Dependencies Only xvi
   Parallelism with Control and Data Dependencies xvi
   Dynamic Task Parallelism and Pipelines xvi
   Supporting Material xvii
What Is Not Covered xviii
Goals xviii

Acknowledgments xix

1 Introduction 1
   The Importance of Potential Parallelism 2
   Decomposition, Coordination, and Scalable Sharing 3
      Understanding Tasks 3
      Coordinating Tasks 4
      Scalable Sharing of Data 5
      Design Approaches 6
   Selecting the Right Pattern 7
   A Word About Terminology 7
   The Limits of Parallelism 8
   A Few Tips 10
   Exercises 11
   For More Information 11
7 Pipelines

The Basics
An Example
  Sequential Image Processing
  The Image Pipeline
  Performance Characteristics
Variations
  Canceling a Pipeline
  Handling Pipeline Exceptions
  Load Balancing Using Multiple Producers
  Pipelines and Streams
  Asynchronous Pipelines
Anti-Patterns
  Thread Starvation
  Infinite Blocking Collection Waits
  Forgetting GetConsumingEnumerable()
  Using Other Producer/Consumer Collections
Design Notes
Related Patterns
Exercises
Further Reading

Appendices
A Adapting Object-Oriented Patterns
  Structural Patterns
    Façade
      Example
      Guidelines
    Decorators
      Example
      Guidelines
    Adapters
      Example
      Guidelines
    Repositories And Parallel Data Access
      Example
      Guidelines
  Singletons and Service Locators
    Implementing a Singleton with the Lazy<T> Class
    Notes
    Guidelines