Microsoft

CLR via C#

Fourth Edition

Developer Reference

Jeffrey Richter

Wintellect®
Know how.

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2012 by Jeffrey Richter

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2012951989
ISBN: 978-0-7356-6745-7

Printed and bound in the United States of America.

Third Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

Microsoft and the trademarks listed at http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Devon Musgrave
Developmental Editor: Devon Musgrave
Project Editor: Carol Dillingham
Editorial Production: Online Training Solutions, Inc.
Technical Reviewer: Christophe Nasarre; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Candace Sinclair
Indexer: Jan Bednarczuk
Cover: Twist Creative • Seattle

[LSI]
[2013-05-17]

CLR via C#

Table of Contents

- Cover
- Copyright
- Dedication
- Contents at a Glance
- Table of Contents
- Foreword
- Introduction
    - Who This Book Is For
    - Acknowledgments
    - Errata & Book Support
    - We Want to Hear from You
    - Stay in Touch
- Part I: CLR Basics
    - Chapter 1: The CLRs Execution Model
        - Compiling Source Code into Managed Modules
        - Combining Managed Modules into Assemblies
        - Loading the Common Language Runtime
        - Executing Your Assemblys Code
            - IL and Verification
            - Unsafe Code
        - The Native Code Generator Tool: NGen.exe
        - The Framework Class Library
        - The Common Type System

Pearson

Table of Contents

The Common Language Specification

Interoperability with Unmanaged Code

Chapter 2: Building, Packaging, Deploying, and Administering Applications and Types

.NET Framework Deployment Goals

Building Types into a Module

Response Files

A Brief Look at Metadata

Combining Modules to Form an Assembly

Adding Assemblies to a Project by Using the Visual Studio IDE

Using the Assembly Linker

Adding Resource Files to an Assembly

Assembly Version Resource Information

Version Numbers

Culture

Simple Application Deployment (Privately Deployed Assemblies)

Simple Administrative Control (Configuration)

Chapter 3: Shared Assemblies and Strongly Named Assemblies

Two Kinds of Assemblies, Two Kinds of Deployment

Giving an Assembly a Strong Name

The Global Assembly Cache

Building an Assembly That References a Strongly Named Assembly

Strongly Named Assemblies Are Tamper-Resistant

Delayed Signing

Privately Deploying Strongly Named Assemblies

How the Runtime Resolves Type References

Advanced Administrative Control (Configuration)

Publisher Policy Control

Part II: Designing Types

Pearson

Table of Contents

Chapter 4: Type Fundamentals

- All Types Are Derived from System.Object
- Casting Between Types
    - Casting with the C# is and as Operators
- Namespaces and Assemblies
- How Things Relate at Run Time

Chapter 5: Primitive, Reference, and Value Types

- Programming Language Primitive Types
    - Checked and Unchecked Primitive Type Operations
- Reference Types and Value Types
- Boxing and Unboxing Value Types
    - Changing Fields in a Boxed Value Type by Using Interfaces (and Why You Shouldnt Do This)
    - Object Equality and Identity
- Object Hash Codes
- The dynamic Primitive Type

Chapter 6: Type and Member Basics

- The Different Kinds of Type Members
- Type Visibility
    - Friend Assemblies
- Member Accessibility
- Static Classes
- Partial Classes, Structures, and Interfaces
- Components, Polymorphism, and Versioning
    - How the CLR Calls Virtual Methods, Properties, and Events
    - Using Type Visibility and Member Accessibility Intelligently
    - Dealing with Virtual Methods When Versioning Types

Chapter 7: Constants and Fields

- Constants
- Fields

Pearson

Table of Contents

Chapter 8: Methods

- Instance Constructors and Classes (Reference Types)
- Instance Constructors and Structures (Value Types)
- Type Constructors
- Operator Overload Methods
  - Operators and Programming Language Interoperability
- Conversion Operator Methods
- Extension Methods
  - Rules and Guidelines
  - Extending Various Types with Extension Methods
  - The Extension Attribute
- Partial Methods
  - Rules and Guidelines

Chapter 9: Parameters

- Optional and Named Arguments
  - Rules and Guidelines
  - The DefaultParameterValue and Optional Attributes
- Implicitly Typed Local Variables
- Passing Parameters by Reference to a Method
- Passing a Variable Number of Arguments to a Method
- Parameter and Return Type Guidelines
- Const-ness

Chapter 10: Properties

- Parameterless Properties
  - Automatically Implemented Properties
  - Defining Properties Intelligently
  - Object and Collection Initializers
  - Anonymous Types
  - The System.Tuple Type
- Parameterful Properties

Pearson

Table of Contents

The Performance of Calling Property Accessor Methods

Property Accessor Accessibility

Generic Property Accessor Methods

Chapter 11: Events

Designing a Type That Exposes an Event

Step #1: Define a type that will hold any additional information that should be sent to receivers of the event notification

Step #2: Define the event member

Step #3: Define a method responsible for raising the event to notify registered objects that the event has occurred

Step #4: Define a method that translates the input into the desired event

How the Compiler Implements an Event

Designing a Type That Listens for an Event

Explicitly Implementing an Event

Chapter 12: Generics

Generics in the Framework Class Library

Generics Infrastructure

Open and Closed Types

Generic Types and Inheritance

Generic Type Identity

Code Explosion

Generic Interfaces

Generic Delegates

Delegate and Interface Contra-variant and Covariant Generic Type Arguments

Generic Methods

Generic Methods and Type Inference

Generics and Other Members

Verifiability and Constraints

Primary Constraints

Secondary Constraints

Constructor Constraints

Pearson

Table of Contents

Other Verifiability Issues

Chapter 13: Interfaces

Class and Interface Inheritance

Defining an Interface

Inheriting an Interface

More About Calling Interface Methods

Implicit and Explicit Interface Method Implementations (Whats Happening Behind the Scenes)

Generic Interfaces

Generics and Interface Constraints

Implementing Multiple Interfaces That Have the Same Method Name and Signature

Improving Compile-Time Type Safety with Explicit Interface Method Implementations

Be Careful with Explicit Interface Method Implementations

Design: Base Class or Interface?

Part III: Essential Types

Chapter 14: Chars, Strings, and Working

Characters

The System.String Type

Constructing Strings

Strings Are Immutable

Comparing Strings

String Interning

String Pooling

Examining a Strings Characters and Text Elements

Other String Operations

Constructing a String Efficiently

Constructing a StringBuilder Object

StringBuilder Members

Pearson

Table of Contents

- Obtaining a String Representation of an Object: ToString
  - Specific Formats and Cultures
  - Formatting Multiple Objects into a Single String
  - Providing Your Own Custom Formatter
- Parsing a String to Obtain an Object: Parse
- Encodings: Converting Between Characters and Bytes
  - Encoding and Decoding Streams of Characters and Bytes
  - Base-64 String Encoding and Decoding
- Secure Strings

Chapter 15: Enumerated Types and Bit Flags

- Enumerated Types
- Bit Flags
- Adding Methods to Enumerated Types

Chapter 16: Arrays

- Initializing Array Elements
- Casting Arrays
- All Arrays Are Implicitly Derived from System.Array
- All Arrays Implicitly Implement IEnumerable, ICollection, and IList
- Passing and Returning Arrays
- Creating Non-Zero Lower Bound Arrays
- Array Internals
- Unsafe Array Access and Fixed-Size Array

Chapter 17: Delegates

- A First Look at Delegates
- Using Delegates to Call Back Static Methods
- Using Delegates to Call Back Instance Methods
- Demystifying Delegates
- Using Delegates to Call Back Many Methods (Chaining)
  - C#s Support for Delegate Chains

Pearson

Table of Contents

Having More Control over Delegate Chain Invocation

Enough with the Delegate Definitions Already (Generic Delegates)

C#s Syntactical Sugar for Delegates

Syntactical Shortcut #1: No Need to Construct a Delegate Object

Syntactical Shortcut #2: No Need to Define a Callback Method (Lambda Expressions)

Syntactical Shortcut #3: No Need to Wrap Local Variables in a Class Manually to Pass Them to a Callback Method

Delegates and Reflection

Chapter 18: Custom Attributes

Using Custom Attributes

Defining Your Own Attribute Class

Attribute Constructor and Field/Property Data Types

Detecting the Use of a Custom Attribute

Matching Two Attribute Instances Against Each Other

Detecting the Use of a Custom Attribute Without Creating Attribute-Derived Objects

Conditional Attribute Classes

Chapter 19: Nullable Value Types

C#s Support for Nullable Value Types

C#s Null-Coalescing Operator

The CLR Has Special Support for Nullable Value Types

Boxing Nullable Value Types

Unboxing Nullable Value Types

Calling GetType via a Nullable Value Type

Calling Interface Methods via a Nullable Value Type

Part IV: Core Facilities

Chapter 20: Exceptions and State Management

Defining Exception

Exception-Handling Mechanics

The try Block

The catch Block

Table of Contents

The finally Block

The System.Exception Class

FCL-Defined Exception Classes

Throwing an Exception

Defining Your Own Exception Class

Trading Reliability for Productivity

Guidelines and Best Practices

Use finally Blocks Liberally

Dont Catch Everything

Recovering Gracefully from an Exception

Backing Out of a Partially Completed Operation When an Unrecoverable Exception OccursMaintaining State

Hiding an Implementation Detail to Maintain a Contract

Unhandled Exceptions

Debugging Exceptions

Exception-Handling Performance Considerations

Constrained Execution Regions (CERs)

Code Contracts

Chapter 21: The Managed Heap and Garbage Collection

Managed Heap Basics

Allocating Resources from the Managed Heap

The Garbage Collection Algorithm

Garbage Collections and Debugging

Generations: Improving Performance

Garbage Collection Triggers

Large Objects

Garbage Collection Modes

Forcing Garbage Collections

Monitoring Your Applications Memory Usage

Working with Types Requiring Special Cleanup

Using a Type That Wraps a Native Resource

Table of Contents

An Interesting Dependency Issue
Other GC Features for Use with Native Resources
Finalization Internals
Monitoring and Controlling the Lifetime of Objects Manually
Chapter 22: CLR Hosting and AppDomains
CLR Hosting
AppDomains
Accessing Objects Across AppDomain Boundaries
AppDomain Unloading
AppDomain Monitoring
AppDomain First-Chance Exception Notifications
How Hosts Use AppDomains
Executable Applications
Microsoft Silverlight Rich Internet Applications
Microsoft ASP.NET and XML Web Services Applications
Microsoft SQL Server
Your Own Imagination
Advanced Host Control
Managing the CLR by Using Managed Code
Writing a Robust Host Application
How a Host Gets Its Thread Back
Chapter 23: Assembly Loading and Reflection
Assembly Loading
Using Reflection to Build a Dynamically Extensible Application
Reflection Performance
Discovering Types Defined in an Assembly
What Exactly Is a Type Object?
Building a Hierarchy of Exception-Derived Types
Constructing an Instance of a Type
Designing an Application That Supports Add-Ins
Using Reflection to Discover a Types Members

Table of Contents

Discovering a Types Members

Invoking a Types Members

Using Binding Handles to Reduce Your Processs Memory Consumption

Chapter 24: Runtime Serialization

Serialization/Deserialization Quick Start

Making a Type Serializable

Controlling Serialization and Deserialization

How Formatters Serialize Type Instances

Controlling the Serialized/Deserialized Data

How to Define a Type That Implements ISerializable When the Base Type Doesnt Implement This Interface

Streaming Contexts

Serializing a Type As a Different Type and Deserializing an Object As a Different Object

Serialization Surrogates

Surrogate Selector Chains

Overriding the Assembly and/or Type When Deserializing an Object

Chapter 25: Interoperating with WinRT Components

CLR Projections and WinRT Component Type System Rules

WinRT Type System Core Concepts

Framework Projections

Calling Asynchronous WinRT APIs from .NET Code

Interoperating Between WinRT Streams and .NET Streams

Passing Blocks of Data Between the CLR and WinRT

Defining WinRT Components in C#

Part V: Threading

Chapter 26: Thread Basics

Why Does Windows Support Threads?

Thread Overhead

Stop the Madness

CPU Trends

Pearson

Table of Contents

CLR Threads and Windows Threads

Using a Dedicated Thread to Perform an Asynchronous Compute-Bound Operation

Reasons to Use Threads

Thread Scheduling and Priorities

Foreground Threads vs. Background Threads

What Now?

Chapter 27: Compute-Bound Asynchronous Operations

Introducing the CLRs Thread Pool

Performing a Simple Compute-Bound Operation

Execution Contexts

Cooperative Cancellation and Timeout

Tasks

Waiting for a Task to Complete and Getting Its Result

Canceling a Task

Starting a New Task Automatically When Another Task Completes

A Task May Start Child Tasks

Inside a Task

Task Factories

Task Schedulers

Parallels Static For, ForEach, and Invoke Methods

Parallel Language Integrated Query

Performing a Periodic Compute-Bound Operation

So Many Timers, So Little Time

How the Thread Pool Manages Its Threads

Setting Thread Pool Limits

How Worker Threads Are Managed

Chapter 28: I/O-Bound Asynchronous Operations

How Windows Performs I/O Operations

C#s Asynchronous Functions

How the Compiler Transforms an Async Function into a State Machine

Table of Contents

Async Function Extensibility

Async Functions and Event Handlers

Async Functions in the Framework Class Library

Async Functions and Exception Handling

Other Async Function Features

Applications and Their Threading Models

Implementing a Server Asynchronously

Canceling I/O Operations

Some I/O Operations Must Be Done Synchronously

FileStream-Specific Issues

I/O Request Priorities

Chapter 29: Primitive Thread Synchronization Constructs

Class Libraries and Thread Safety

Primitive User-Mode and Kernel-Mode Constructs

User-Mode Constructs

Volatile Constructs

Interlocked Constructs

Implementing a Simple Spin Lock

The Interlocked Anything Pattern

Kernel-Mode Constructs

Event Constructs

Semaphore Constructs

Mutex Constructs

Chapter 30: Hybrid Thread Synchronization Constructs

A Simple Hybrid Lock

Spinning, Thread Ownership, and Recursion

Hybrid Constructs in the Framework Class Library

The ManualResetEventSlim and SemaphoreSlim Classes

The Monitor Class and Sync Blocks

The ReaderWriterLockSlim Class

Pearson

Table of Contents

The OneManyLock Class
The CountdownEvent Class
The Barrier Class
Thread Synchronization Construct Summary
The Famous Double-Check Locking Technique
The Condition Variable Pattern
Asynchronous Synchronization
The Concurrent Collection Classes
Index
About the Author
Survey

Pearson