Contents

About the Author ..... *xiii*
About the Technical Reviewer ..... *xiv*
Acknowledgments ..... *xv*
Introduction ..... *xvii*

Chapter 1 Language Interoperability 1

Language Interoperability in Action 1
Language Interoperability in Theory 8
Language Interoperability in Practice 12
Conclusion 21

Chapter 2 ilasm Directives and Attributes 23

General Assembly ilasm 23
Defining the Assembly 23
Defining the Modules 30
Defining the UI 30
Concurrent Assembly Execution 31
Defining Data Segments 31
Type-Specific Directives 32
Defining the Type 32
Adding Fields to a Type 41
Field Scope 42
Field Initialization 43
serializable Redux 44
Adding Methods to Types 45
Type and Instance Methods 45
Method Implementation Details 46
Calling Conventions 46
Passing Arguments 48
Unmanaged Calls 48
Type and Instance Initializers 49
beforefieldinit Redux 50
Exporting Methods 51
Method Body Directives 53
Defining Properties in Types 62
Adding Events to Types 64
Conclusion 67

Chapter 3 CIL Opcodes 69

Basic Stack Information 69
Opcode Essentials 70
Basic Opcodes 71
- Using Local Variables 71
- Conversions 74
- Using Method Arguments 77
- Using Type Fields 79

Basic Operations 80
- Arithmetic Operations 80
- Binary Operations 83
- Comparison Operations 85

Object Instructions 86
- Creating Type Instances 86
- Calling Methods 88
- Constructors 91
- Unmanaged Method Calls 91
- Method Pointers 93
- Boxing, Unboxing, and Value Type Indirection 94
- Type Casting 96
- Array Manipulation 97
- Memory Manipulation 99
- String Manipulation 101

Controlling Code Flow 102
Handling Exceptions 106
Opcode Miscellanea 114
- Variable Argument Lists 114
- .maxstack Calculations 116
- Pointer Prefixes 117
- Method Jumps 117

Conclusion 119

Chapter 4 ilasm and CIL in Practice 121

Generating Random Numbers 121
Designing the Essentials 122
Implementing the Essentials 123
- Stubbing Out the Delegate 123
- Field Definitions 125
- Event Methods 125

Constructors 127
Creating the Properties 128
Implementing Generate() 129
Implementing GenerateViaDelegate() 131
Implementing GenerateViaEvent() 132
Generating the RIG Assembly 133
Designing the Test Harness 133
Implementing the Test Harness 134
Referencing Assemblies for UI Design 134
Defining UI Components 135
Initializing the Form 136
Calling Generate() 138
Calling GenerateViaDelegate() 141
Calling GenerateViaEvent() 143
Generating the RIGTestClient Assembly 144
Conclusion 147

Chapter 5 Debugging CIL 149

Debug Builds 149
The Command-Line Tool: cordbg.exe 151
cordbg Basics 151
Watching Register Values and Native Instructions 156
Setting Breakpoints 159
Modes 160
Recording and Using cordbg Scripts 161
The GUI Tool: dbgclr.exe 163
Debugging Windows 165
Saving the Solution 170
Debugger Differences 171
Debugging In-Process Assemblies 171
DLLs in dbgclr 171
DLLs in cordbg 176
Registry Control 176
Verifying Assemblies 178
Debugging Compiled Programs 182
Preventing Recompilation 186
Conclusion 192

Chapter 6 .NET Languages and CIL 193

Debug and Release Builds 193
The C# Implementation 194
The VB .NET Implementation 198
The Component Pascal Implementation 202
Commentary 204
Language Constructs 205
VB .NET's With Statement 205
Implementing Interface Methods 208
On Error Resume Next, or How to Create a Lot of CIL 214
Active Objects 221
Language Interoperability: The Real Story 224
Inheritance with Oberon .NET Types 224
Overloaded and Overridden Methods 226
The Other Property 230
Overloading Methods in CIL 234
Conclusion 237

Chapter 7 Emitting Types 239

Applying CIL 239
Emitter Basics 240
Defining the Essentials 240
Creating Assemblies 241
Building Types 243
Adding Fields 244
Emitting Methods 244
Adding Methods 244
Implementing Methods 246
Exception Handling 250
Persisting the Results 251
Baking Types 251
Transient-Only Assemblies 252
Persistent-Only Assemblies 252
Transient, Persistent Assemblies 254
Beyond the Basics 254
Setting Entry Points 254
Adding Attributes 255
Modifying Method Parameter Information 256

Debugging Dynamic Assemblies *258*
Emitting Debug Information *258*
Adding to the Symbol Writer *258*
Reloading the Assembly *260*
Where's the Target IL File? *262*
Creating the DebugFile Class *263*
Debugging Transient Modules *269*
Conclusion *270*

Chapter 8 Dynamic Proxies in .NET *271*

Separating Concerns *271*
Functional Specifications *274*
Class Design and Method Call Conditions *275*
Receiving a Notification before Method Invocation *276*
Receiving a Notification after Method Invocation *277*
When Method Invocations Go Bad *278*
Proxy Call Flow *278*
Implementing Proxy Creation *279*
Preconditional Checks *279*
Caching Types, Modules, and Assemblies *282*
Defining the Constructors *285*
Implementing the Interfaces *287*
Invoking Target Methods *287*
Setting Up the Method Hooks *294*
Testing the Solution *301*
Loading the Form *302*
Creating Proxies *302*
Type Inspection *305*
Contexts and .NET *309*
Dynamic Proxies and Contexts *314*
Conclusion *315*

Chapter 9 CIL Tips *317*

Be CLS-Compliant Aware *317*
Use Verification Tools *317*
Comments and Code Regions *318*
Stay Location Savvy When Emitting Assemblies *318*
Final Thoughts *318*

Appendix A The Future of CIL *319*

Generics in CIL *319*
Extended IL *322*
Inline CIL *324*

Appendix B .NET Languages *327*

Index *329*