

Contents

About the Author	xix
About the Technical Reviewers	xxi
Acknowledgments	xxii
Introduction	xxv

PART 1 ■ ■ ■ Quick Start

■ CHAPTER 1	Simple Sample	3
	Basics of the Common Language Runtime	3
	Simple Sample: The Code	7
	Program Header	8
	Class Declaration	9
	Field Declaration	11
	Method Declaration	12
	Global Items	16
	Mapped Fields	17
	Data Declaration	18
	Value Type As Placeholder	19
	Calling Unmanaged Code	19
	Forward Declaration of Classes	21
	Summary	22
■ CHAPTER 2	Enhancing the Code	23
	Compacting the Code	23
	Protecting the Code	26
	Summary	30
■ CHAPTER 3	Making the Coding Easier	31
	Aliasing	31
	Compilation Control Directives	34
	Referencing the Current Class and Its Relatives	37
	Summary	38

PART 2 ■ ■ ■ Underlying Structures

■ CHAPTER 4	The Structure of a Managed Executable File	41
	PE/COFF Headers	42
	MS-DOS Header/Stub and PE Signature	42
	COFF Header	43
	PE Header	47
	Section Headers	53
	Common Language Runtime Header	55
	Header Structure	55
	Flags Field	57
	EntryPointToken Field	58
	VTableFixups Field	58
	StrongNameSignature Field	59
	Relocation Section	59
	Text Section	61
	Data Sections	63
	Data Constants	63
	V-Table	63
	Unmanaged Export Table	64
	Thread Local Storage	66
	Resources	67
	Unmanaged Resources	67
	Managed Resources	69
	Summary	70
	Phase 1: Initialization	70
	Phase 2: Source Code Parsing	70
	Phase 3: Image Generation	70
	Phase 4: Completion	71
■ CHAPTER 5	Metadata Tables Organization	73
	What Is Metadata?	73
	Heaps and Tables	75
	Heaps	75
	General Metadata Header	76
	Metadata Table Streams	79
	RIDs and Tokens	83
	RIDs	83
	Tokens	83

Coded Tokens	85
Metadata Validation	88
Summary	89

PART 3 ■ ■ ■ Fundamental Components

CHAPTER 6	Modules and Assemblies	93
	What Is an Assembly?	93
	Private and Shared Assemblies	93
	Application Domains As Logical Units of Execution	94
	Manifest	96
	Assembly Metadata Table and Declaration	97
	AssemblyRef Metadata Table and Declaration	99
	Autodetection of Referenced Assemblies	101
	The Loader in Search of Assemblies	101
	Module Metadata Table and Declaration	105
	ModuleRef Metadata Table and Declaration	105
	File Metadata Table and Declaration	106
	Managed Resource Metadata and Declaration	107
	ExportedType Metadata Table and Declaration	110
	Order of Manifest Declarations in ILAsm	112
	Single-Module and Multimodule Assemblies	112
	Summary of Metadata Validity Rules	113
	Assembly Table Validity Rules	114
	AssemblyRef Table Validity Rules	114
	Module Table Validity Rules	114
	ModuleRef Table Validity Rules	115
	File Table Validity Rules	115
	ManifestResource Table Validity Rules	115
	ExportedType Table Validity Rules	116
CHAPTER 7	Namespaces and Classes	117
	Class Metadata	118
	TypeDef Metadata Table	120
	TypeRef Metadata Table	120
	InterfaceImpl Metadata Table	121
	NestedClass Metadata Table	121
	ClassLayout Metadata Table	121

Namespace and Full Class Name	122
ILAsm Naming Conventions	122
Namespaces	124
Full Class Names	125
Class Attributes	126
Flags	126
Class Visibility and Friend Assemblies	128
Class References	129
Parent of the Type	129
Interface Implementations	130
Class Layout Information	131
Interfaces	131
Value Types	133
Boxed and Unboxed Values	133
Instance Members of Value Types	134
Derivation of Value Types	135
Enumerations	135
Delegates	136
Nested Types	138
Class Augmentation	140
Summary of the Metadata Validity Rules	142
TypeDef Table Validity Rules	142
Enumeration-Specific Validity Rules	143
TypeRef Table Validity Rules	143
InterfacelImpl Table Validity Rules	144
NestedClass Table Validity Rules	144
ClassLayout Table Validity Rules	144

■ CHAPTER 8 **Primitive Types and Signatures** 145

Primitive Types in the Common Language Runtime	145
Primitive Data Types	145
Data Pointer Types	146
Function Pointer Types	148
Vectors and Arrays	149
Modifiers	151
Native Types	153
Variant Types	155
Representing Classes in Signatures	157
Signatures	158
Calling Conventions	158

Field Signatures	159
Method and Property Signatures	159
MemberRef Signatures	160
Indirect Call Signatures	161
Local Variables Signatures	161
Type Specifications	162
Summary of Signature Validity Rules	163
 CHAPTER 9 Fields and Data Constants	165
Field Metadata	165
Defining a Field	166
Referencing a Field	168
Instance and Static Fields	168
Default Values	169
Mapped Fields	171
Data Constants Declaration	173
Explicit Layouts and Union Declaration	175
Global Fields	177
Constructors vs. Data Constants	179
Summary of Metadata Validity Rules	181
Field Table Validity Rules	181
FieldLayout Table Validity Rules	182
FieldRVA Table Validity Rules	182
FieldMarshal Table Validity Rules	183
Constant Table Validity Rules	183
MemberRef Table Validity Rules	183
 CHAPTER 10 Methods	185
Method Metadata	185
Method Table Record Entries	186
Method Flags	187
Method Name	190
Method Implementation Flags	190
Method Parameters	191
Referencing the Methods	193
Method Implementation Metadata	194
Static, Instance, Virtual Methods	194
Explicit Method Overriding	199
Method Overriding and Accessibility	205

Method Header Attributes	205
Local Variables	207
Class Constructors	209
Class Constructors and the beforefieldinit Flag	210
Module Constructors	212
Instance Constructors	213
Instance Finalizers	215
Variable Argument Lists	216
Method Overloading	218
Global Methods	220
Summary of Metadata Validity Rules	221
Method Table Validity Rules	221
Param Table Validity Rules	223
MethodImpl Table Validity Rules	223
CHAPTER 11 Generic Types	225
Generic Type Metadata	226
GenericParam Metadata Table	228
GenericParamConstraint Metadata Table	229
TypeSpec Metadata Table	229
Constraint Flags	229
Defining Generic Types in ILAsm	230
Addressing the Type Parameters	231
Generic Type Instantiations	232
Defining Generic Types: Inheritance, Implementation, Constraints	233
Defining Generic Types: Cyclic Dependencies	234
The Members of Generic Types	237
Virtual Methods in Generic Types	239
Nested Generic Types	243
Summary of the Metadata Validity Rules	245
CHAPTER 12 Generic Methods	247
Generic Method Metadata	247
MethodSpec Metadata Table	249
Signatures of Generic Methods	249
Defining Generic Methods in ILAsm	250
Calling Generic Methods	251
Overriding Virtual Generic Methods	253
Summary of the Metadata Validity Rules	257

PART 4 ■ ■ ■ Inside the Execution Engine

■ CHAPTER 13	IL Instructions	261
	Long-Parameter and Short-Parameter Instructions	262
	Labels and Flow Control Instructions	263
	Unconditional Branching Instructions	263
	Conditional Branching Instructions	264
	Comparative Branching Instructions	264
	The switch Instruction	265
	The break Instruction	266
	Managed EH Block Exiting Instructions	266
	EH Block Ending Instructions	266
	The ret Instruction	267
	Arithmetical Instructions	267
	Stack Manipulation	267
	Constant Loading	268
	Indirect Loading	269
	Indirect Storing	269
	Arithmetical Operations	270
	Overflow Arithmetical Operations	271
	Bitwise Operations	272
	Shift Operations	273
	Conversion Operations	273
	Overflow Conversion Operations	274
	Logical Condition Check Instructions	275
	Block Operations	276
	Addressing Arguments and Local Variables	276
	Method Argument Loading	277
	Method Argument Address Loading	277
	Method Argument Storing	277
	Method Argument List	278
	Local Variable Loading	278
	Local Variable Reference Loading	278
	Local Variable Storing	278
	Local Block Allocation	279
	Prefix Instructions	279
	Addressing Fields	280
	Calling Methods	281
	Direct Calls	281

Indirect Calls	283
Tail Calls	283
Constrained Virtual Calls	284
Addressing Classes and Value Types	285
Vector Instructions	289
Vector Creation	289
Element Address Loading	290
Element Loading	290
Element Storing	291
Code Verifiability	292

■ CHAPTER 14 Managed Exception Handling 295

EH Clause Internal Representation	295
Types of EH Clauses	297
Label Form of EH Clause Declaration	299
Scope Form of EH Clause Declaration	301
Processing the Exceptions	304
Exception Types	305
Loader Exceptions	306
JIT Compiler Exceptions	306
Execution Engine Exceptions	306
Interoperability Exceptions	308
Subclassing the Exceptions	308
Unmanaged Exception Mapping	309
Summary of EH Clause Structuring Rules	309

PART 5 ■ ■ ■ Special Components

■ CHAPTER 15 Events and Properties 313

Events and Delegates	313
Event Metadata	316
The Event Table	316
The EventMap Table	317
The MethodSemantics Table	317
Event Declaration	318
Property Metadata	321
The Property Table	322
The PropertyMap Table	322

Property Declaration	323
Summary of Metadata Validity Rules	324
Event Table Validity Rules	324
EventMap Table Validity Rules	325
Property Table Validity Rules	325
PropertyMap Table Validity Rules	325
MethodSemantics Table Validity Rules	325
CHAPTER 16 Custom Attributes	327
Concept of a Custom Attribute	327
CustomAttribute Metadata Table	328
Custom Attribute Value Encoding	329
Verbal Description of Custom Attribute Value	331
Custom Attribute Declaration	332
Classification of Custom Attributes	336
Execution Engine and JIT Compiler	337
Interoperation Subsystem	338
Security	340
Remoting Subsystem	341
Visual Studio Debugger	342
Assembly Linker	343
Common Language Specification (CLS) Compliance	344
Pseudocustom Attributes	344
Summary of Metadata Validity Rules	346
CHAPTER 17 Security Attributes	347
Declarative Security	348
Declarative Actions	348
Security Permissions	350
Access Permissions	350
Identity Permissions	354
Custom Permissions	356
Permission Sets	358
Declarative Security Metadata	358
Permission Set Blob Encoding	359
Security Attribute Declaration	360
Summary of Metadata Validity Rules	361

CHAPTER 18	Managed and Unmanaged Code Interoperation	363
	Thunks and Wrappers	364
	P/Invoke Thunks	364
	Implementation Map Metadata	366
	IJW Thunks	367
	COM Callable Wrappers	368
	Runtime Callable Wrappers	369
	Data Marshaling	370
	Blittable Types	371
	In/Out Parameters	371
	String Marshaling	372
	Object Marshaling	373
	More Object Marshaling	375
	Array Marshaling	376
	Delegate Marshaling	376
	Providing Managed Methods As Callbacks for Unmanaged Code	377
	Managed Methods As Unmanaged Exports	380
	Export Table Group	381
	Summary	387
 CHAPTER 19	 Multilanguage Projects	 389
	IL Disassembler	389
	Principles of Round-Tripping	394
	Creative Round-Tripping	395
	Using Class Augmentation	396
	Module Linking Through Round-Tripping	397
	ASMMETA: Resolving Circular Dependencies	398
	IL Inlining in High-Level Languages	400
	Compiling in Debug Mode	402
	Summary	408

PART 6 ■ ■ ■ Appendixes

■ APPENDIX A	ILAsm Grammar Reference	411
	Lexical Tokens	411
	Auxiliary Lexical Tokens	411
	Data Type Nonterminals	411
	Identifier Nonterminals	412
	Class Referencing	412
	Module-Level Declarations	412
	Compilation Control Directives	413
	Module Parameter Declaration	413
	V-Table Fixup Table Declaration	413
	Manifest Declarations	414
	Managed Types in Signatures	416
	Native Types in Marshaling Signatures	417
	Method and Field Referencing	419
	Class Declaration	420
	Generic Type Parameters Declaration	421
	Class Body Declarations	421
	Field Declaration	422
	Method Declaration	423
	Method Body Declarations	424
	External Source Directives	425
	Managed Exception Handling Directives	425
	IL Instructions	426
	Event Declaration	426
	Property Declaration	427
	Constant Declarations	427
	Custom Attribute Declarations	429
	Verbal Description of Custom Attribute Initialization Blob	429
	Security Declarations	430
	Aliasing of Types, Methods, Fields, and Custom Attributes	431
	Data Declaration	431
■ APPENDIX B	Metadata Tables Reference	433
■ APPENDIX C	IL Instruction Set Reference	445

- **APPENDIX D IL Assembler and Disassembler**
 - Command-Line Options** 453
 - IL Assembler 453
 - IL Disassembler 456
 - Output Redirection Options 456
 - ILAsm Code-Formatting Options (PE Files Only) 456
 - File Output Options (PE Files Only) 457
 - File or Console Output Options (PE Files Only) 457
 - Metadata Summary Option 458
- **APPENDIX E Offline Verification Tool Reference** 459
 - Error Codes and Messages 461
- **INDEX** 477