

CONTENTS

<i>Foreword</i>	xiii
<i>Preface</i>	xv

PART I FUNDAMENTAL PROGRAMMING ARCHITECTURE 1

CHAPTER ONE

The Road to Distributed COM	3
From OLE to DCOM	3
How DCOM Evolved from RPC	5
Parallel Processing	6
Advantages of Distributed Computing	7
Economics	7
Performance	8
Reliability	8
Incremental Growth	9
Architecture	10
Building Distributed Systems	10
COM Background	11
Object-Oriented Programming	11
Code Sharing and Reuse	16
Three Faces of COM	18
COM as a Specification	18
COM as a Philosophy	18
COM as a Binary Standard	19
Componentware	20
COM Interfaces	22
Types of Components	24
In-Process	24
Local	25
Remote	25
The COM Library	26
COM as a Foundation	27

Persistent Storage	27
Monikers	27
Uniform Data Transfer	28
ActiveX on COM	28
CHAPTER TWO	
<i>IUnknown</i> and <i>IClassFactory</i>	31
The Interface Definition Language	33
The Component's Client	37
The <i>CoInitialize</i> Function	38
The <i>CoCreateInstance</i> Function	38
The <i>IUnknown</i> Interface Methods	40
The <i>CoUninitialize</i> Function	43
The V-Table Situation	44
Building the Client Project	45
The Problems with <i>QueryInterface</i>	47
The Component	48
The <i>AddRef</i> Method	50
The <i>Release</i> Method	51
The <i>QueryInterface</i> Method:	
Preserving the Identity of a COM Object	51
The <i>ISum::Sum</i> Method (Finally)	53
<i>CoCreateInstance</i> (Again)	54
The <i>IClassFactory</i> Interface	54
Exported DLL Functions	58
<i>CoCreateInstance</i> (The Truth)	60
Building the Component Project	63
Component Self-Registration	67
COM Reuse Mechanisms	72
Containment	72
Aggregation	73
CHAPTER THREE	
Type Libraries and Language Integration	79
Eureka! Type Libraries	80
Using Type Libraries	80
Creating a Type Library	81
Registering the Type Library	83

Embedding the Type Library in the Component	86
An Easy C++ Client That Utilizes the Type Library	88
C++ Templates (A Quick Introduction)	89
Smart Pointers	92
Namespaces	94
Active Template Library	96
The ATL COM AppWizard	97
The ATL Object Wizard	97
Adding Methods and Properties to an Interface Using ATL	99
Building a Simple COM Object Using ATL	101
COM Programming in Visual Basic	102
Pushing the Limits of Visual Basic	105
Building a Client in Visual Basic	110
Building a Component in Visual Basic	111
COM Programming in Java	112
The Java Type Library Wizard	114
Calling a COM Object from Java	116
Implementing COM Objects in Java	117
The ActiveX Component Wizard for Java	118
Building a Client in Java	121
Building a Component in Java	122
ActiveX Controls and JavaBeans Integration	124
Trusted vs. Untrusted Applets	125

CHAPTER FOUR

Threading Models and Apartments 127	
Spinning a Yarn: A Quick Review of Threads	128
Apartments	129
The Single-Threaded Apartment Model	132
The Multithreaded Apartment Model	138
Have We Got a Threading Model for You	140
Apartment Interactions	144
In-Process Interactions	145
Aggregating with the Free-Threaded Marshaler	152
Implementing Multithreaded Local Components	159
Local Component Shutdown	161
The Ten Threading Commandments	164

PART II COM FACILITIES 167

CHAPTER FIVE

Automation and Component Categories 169

The *IDispatch* Interface 170

 Accessing COM Services 172

 Implementing *IDispatch* 176

Building an Automation Client in C++ 184

Building an Automation Client in Visual Basic 187

Building an Automation Client in VBScript 190

Scriptlets: Building COM Objects in HTML 192

Error Handling 193

 Rich Error Information 195

Component Categories 200

 Category Identifiers 202

CHAPTER SIX

Connection Points and Type Information 211

A Simple Version of a Connectable Object 212

 The *IOutgoing* Interface 212

 The *IConnectionPoint* Interface 214

 The *IConnectionPointContainer* Interface 216

 Implementing a Sink in C++ 219

 A Visual Basic Sink 223

 A Java Sink 230

A Complete Implementation of a Connectable Object 231

 Enumerators 232

 When to Use Connection Points 238

Type Information 239

 Creating a Type Library at Run Time 240

 Obtaining Type Information 248

CHAPTER SEVEN

Monikers and Structured Storage 251

Initializing Objects 251

 Class Objects 253

 Custom Activation Interfaces 253

Monikers (One More Time)	254
The <i>IMoniker</i> Interface	256
The <i>MkParseDisplayName</i> Function	258
The Class Moniker	261
Managing the Lifetime of an Executable Component: <i>IExternalConnection</i>	264
Marvelous Moniker: Improving the Class Moniker	267
The Java Moniker	274
The Running Object Table	275
Structured Storage	277
The <i>IStorage</i> Interface	279
The <i>IStream</i> Interface	281
The <i>IPropertySetStorage</i> and <i>IPropertyStorage</i> Interfaces	282
 PART III REMOTING ARCHITECTURE	 287
CHAPTER EIGHT	
DLL Surrogates and Executable Components	289
DLL Surrogates	290
Running In-Process Components Locally	290
Running Components Remotely	293
Writing a Custom Surrogate	295
Marshaling	298
Standard Marshaling	300
Type Library Marshaling	302
Custom Marshaling	304
Executable Components	305
Building an Executable Component	307
Class Emulation	320
The Class Store	321
Building Singletons in COM	322
CHAPTER NINE	
Standard and Custom Marshaling	325
Marshaling Interface Pointers—An Overview	326
Re-Creating an Interface's V-Table	327
Interprocess Communications	327
The Proxy/Stub DLL	329

Will That Be Custom or Standard Marshaling? 329

Standard Marshaling 332

 The Standard Marshaling Interfaces 335

 Registering the Proxy/Stub DLL 348

Handler Marshaling 349

Custom Marshaling 352

 Can You Say “Custom Marshaling”? 353

 Pardon Me, What Is the CLSID of Your Proxy Object? 357

 How Big Did You Say Your Interface Is? 358

 Unmarshaling the Interface Pointer 363

Converting Marshaled Interface Pointers to Strings 368

CHAPTER TEN

The Interface Definition Language 373

Types 374

Directional Attributes 375

Arrays 378

 Fixed Arrays 379

 Conformant Arrays 380

 Varying Arrays 382

 Open Arrays 384

 Character Arrays 385

 Multidimensional Arrays 388

 Passing Arrays of User-Defined Types from Visual Basic 389

Pointers 390

 Full Pointers 391

 Unique Pointers 391

 Reference Pointers 392

 Interface Pointers 393

Interface Design Recommendations 396

CHAPTER ELEVEN

Security 399

Security Models 399

 The Windows NT Security Model 399

 The COM Security Model 401

 Why Does COM Need a Security Model? 402

Declarative Security: The Registry	403
Default Security	404
Configuring Default Access and Launch Permissions	407
Configuring Component Security: The AppID Key	410
The <i>IAccessControl</i> Interface	412
Configuring Component Identity	417
Programmatic Security	421
The <i>CoInitializeSecurity</i> Function	421
Using the <i>IAccessControl</i> Interface with <i>CoInitializeSecurity</i>	426
Activation Credentials: The COAUTHINFO Structure	430
The <i>IServerSecurity</i> Interface	433
The <i>IClientSecurity</i> Interface	436
 CHAPTER TWELVE	
The Network Protocol	439
Spying on the Network Protocol	442
Remote Activation	445
Calling All Remote Objects	447
The ORPCTHIS and ORPCTHAT Structures	450
Marshaled Interface Pointers	451
The Standard Object Reference	455
The DUALSTRINGARRAY Structure	456
The <i>IRemUnknown</i> Interface	458
The <i>IRemUnknown2</i> Interface	462
The OXID Resolver	463
DCOM Garbage Collection	467
A Remote Method Call	470
Channel Hooks	471
A Useful Channel Hook: Obtaining the Client's Name	474
Using <i>ClientChannelHook</i>	477
 CHAPTER THIRTEEN	
Microsoft Transaction Server and COM+	479
Three-Tier Architecture	479
The Client Tier	480
The Business-Rules Tier	480
The Database Server Tier	481
Three-Tier Architecture Concerns	482

Microsoft Transaction Server	482
Designing Components for MTS	483
Just-in-Time Activation	486
State	489
Transactions	495
Security	502
Error Handling	505
Registering Components with MTS	506
OLE, Network OLE, COM, ActiveX, DCOM,...COM+	507
The COM+ Run Time	508
 APPENDIX	
Remote Procedure Calls	513
The Design and Purpose of RPC	514
Interface Definition Language	514
Binding	515
The RPC Charade	516
Handles	516
Subtle Differences Between RPC and LPC	517
Global Variables	517
Passing Pointers	517
Multithreading and Structured Exception Handling	519
The Prime Application	520
Client Initialization	520
Client Computation	523
The Prime Server	524
Context Rundown	525
Debugging RPC Applications	526
The Advantage of Distributed Prime Calculations	526
Index	531