

Contents

<i>Acknowledgments</i>	xxiii
<i>Introduction</i>	xxv

Chapter 1 Understanding Platform

Invocation Services	1
<i>The Two Faces of Unmanaged Code</i>	1
<i>Understanding the C-Style DLL</i>	2
Exporting Custom Members	3
<i>Building a Custom C-Based DLL</i>	5
Functions Using Basic Data Types and Arrays	6
Functions Receiving Structures (and Structures Containing Structures)	7
Functions Using Class Types	8
<i>Viewing Your Imports and Exports Using dumpbin.exe</i>	9
<i>Deploying Traditional DLLs</i>	12
<i>A Dynamic C++ DLL Client</i>	15
Dynamically Loading an External Library	15
Invoking Members	16
<i>The Atoms of PInvoke</i>	18
Data Type Conversions	18
The Marshal Class	20
The DllImportAttribute Type	25
<i>A Trivial PInvoke Example</i>	26
Specifying the ExactSpelling Field	29
Specifying the Character Set	29
Specifying Calling Conventions	30
Specifying Function Entry Points	31
SetLastError and Marshal.GetLastWin32Error()	32
<i>Interacting with MyCustomDLL.dll</i>	33
Marshaling Arrays	34
Passing Structures	35
Receiving Allocated Structures	37
Interacting with Exported Class Types	39
Examining the Underlying IL	41
<i>Working with Unmanaged Callbacks</i>	42
<i>A Simple Callback Example</i>	43
<i>A More Interesting Callback Function</i>	44
<i>Building a C# Callback Client</i>	46
<i>Summary</i>	49

<i>Chapter 2 The Anatomy of a COM Server</i>	51
<i>Of Classes and Interfaces</i>	51
Interfaces from a Non-COM Perspective	52
Cleaning Up the Interface Definition	54
Building the Class Types	55
Interfaces Are Strongly Typed Variables	58
Classes Can Support Multiple Interfaces	59
Interfaces Provide a Versioning Scheme	61
<i>The Composition of a COM DLL</i>	63
<i>The Role of Type Libraries</i>	65
The Full MIDL Output	66
<i>The Role of the System Registry</i>	66
<i>Creating the COM DLL Project Workspace</i>	67
<i>Understanding COM Interfaces</i>	68
The Role of the GUID	68
The Role of IUnknown	71
Defining the ICar Interface in IDL	73
The Role of the HRESULT	74
IDL Parameter Attributes.....	74
Defining the Coclass (ComCar) in IDL.....	75
Defining the [default] Interface	76
Defining and Supporting Multiple Interfaces.....	76
Compiling the IDL File with the MIDL Compiler	77
Examining the MIDL-Generated Files.....	78
<i>A Brief Word on COM-Centric Macros</i>	79
<i>Implementing the ComCar</i>	80
Implementing IUnknown.....	81
Implementing ICar and IRadio	83
<i>Understanding IClassFactory</i>	84
<i>Building Your Class Factory</i>	85
Implementing IClassFactory::CreateInstance()	86
Implementing IClassFactory::LockServer()	87
<i>Implementing DLL Component Housing</i>	88
Managing Server Lifetime: DllCanUnloadNow().....	88
Contending with DllRegisterServer() and DllUnregisterServer()	90
<i>Exporting the Exports</i>	90
<i>Registering the COM Server</i>	91
Programmatic Identifiers (ProgIDs)	91
A Critical Key: HKEY_CLASSES_ROOT \ CLSID	92
Another Critical Key: HKEY_CLASSES_ROOT\TypeLib	94
Other COM Registration Possibilities	94
Registering Your COM Servers.....	95

<i>Developing a C++ COM Client</i>	97
Activating COM Objects	97
Accessing a Coclass Using CoCreateInstance()	100
Building a C++ Client Using the #import Directive	102
A Visual Basic 6.0 COM Client	103
<i>Understanding COM Properties</i>	105
IDL Syntax for Interface Properties	106
Properties: The Client's Point of View	107
<i>Building Scriptable Objects (Using IDispatch)</i>	108
Understanding Dispinterfaces and DISPIDs	109
Obtaining DISPIDs.....	111
Invoking the Member	111
<i>The VARIANT Data Type</i>	112
Working with VARIANTS (in C++)	114
Working with VARIANTS (in VB 6.0)	115
The DISPPARAMS Structure.....	115
<i>A C++ IDispatch Example</i>	116
<i>A Visual Basic IDispatch Client</i>	117
<i>Understanding the Dual Interface</i>	118
<i>Defining a Scriptable Object</i>	118
Implementing IScriptableCar	119
<i>Building the Class Factory</i>	123
<i>Updating DllGetClassObject</i>	123
<i>Updating the Server's Registration File</i>	124
<i>Building a VBScript Late-Bound Client</i>	124
<i>Summary</i>	126

Chapter 3 A Primer on COM

Programming Frameworks

<i>The Role of the Active Template Library</i>	127
Generating the Component Housing	128
The ATL Project Files.....	129
Inserting COM Objects Using the ATL Object Wizard	133
Code Updates	135
ATL's Implementation of Your Coclass.....	136
ATL's Registration Support.....	137
Adding Members to the [Default] Interface	140
Adding a More Interesting Method.....	142
Supporting Additional COM Interfaces.....	143

<i>The Role of Visual Basic 6.0</i>	146
Building COM Servers Using Visual Basic 6.0	146
Defining Auxiliary Interfaces	148
Implementing Interfaces in VB 6.0	149
Setting Binary Compatibility	151
Viewing the Generated IDL Using Oleview.exe	152
Making Use of Your COM Servers	156
<i>Summary</i>	159
 Chapter 4 COM Type Information	161
 <i>The Format of a COM IDL File</i>	162
<i>Defining COM Types</i>	163
<i>Primitive COM Data Types</i>	164
The Oleautomation (aka Variant) Data Types	166
<i>IDL Method Parameter Attributes</i>	167
<i>Defining COM Interface Types in IDL</i>	171
Defining Raw Dispinterfaces	171
Defining Dual Interfaces	172
Building IUnknown-Derived Interfaces	173
Common IDL Interface Modifiers	173
Defining COM Classes in IDL	174
Defining IDL Enumerations	176
Defining IDL Structures	176
COM String Representation	177
COM (Safe)Array Representation	179
COM Interface Types As Method Parameters	184
<i>The ITypeInfo Interface</i>	185
A Brief Word on ITypeInfo2	188
Related ITypeInfo Data Types	188
<i>Generating COM Type Information Programmatically</i>	189
A Brief Word on ICreateTypeLib2 and ICreateTypeInfo2	190
The Target IDL	190
<i>Building the Type Library (ICreateTypeLib)</i>	191
<i>Creating the IHello Interface</i>	193
Breaking Down the CreateInterface() Helper Method	195
<i>Building the SayHello() Method</i>	198
<i>Building the Hello Coclass</i>	200
<i>Testing the Application</i>	201

<i>Programmatically Reading COM Type Information</i>	203
The Program Skeleton.....	204
Displaying COM Library Information	205
Dumping COM Type Information	207
Listing CoClass Statistics	208
Listing IDispatch-Based Interface Statistics	209
Listing IUnknown-Based Interface Statistics	209
Listing COM Enumeration Statistics	209
Reading the MyTypeLib.tlb file	210
<i>Defining Custom IDL Attributes</i>	212
Reading Custom Attributes	214
<i>Introducing the System.Runtime.InteropServices Namespace</i>	218
<i>Building a C# COM Type Information Viewer</i>	220
Loading the COM Type Library.....	221
Loading the COM Type Library.....	222
Displaying the COM Types	224
<i>Summary</i>	228
<i>Chapter 5 The Anatomy of a .NET Server</i>	229
<i>The Philosophy of .NET</i>	230
<i>The Building Blocks of .NET</i>	231
<i>Working with Managed Languages</i>	232
<i>The Composition of a .NET Binary</i>	233
<i>Building a C# Code Library</i>	235
Building the Abstract Car Type.....	238
Building the Derived MiniVan Type.....	240
Implementing the Convertibles.....	240
Establishing the Assembly Manifest.....	242
<i>Introducing ILDasm.exe</i>	243
Viewing the Assembly Manifest	243
Viewing the Type Metadata	245
Viewing (Partial) Metadata for the CarColor Enumeration	246
Viewing (Partial) Metadata for the IConvertible Interface.....	247
Viewing (Partial) Metadata for a Derived Type	247
Remaining Bits of Interesting Metadata.....	248
(Not) Viewing the Underlying IL Code	249
<i>Building a Managed Client</i>	250
Configuring Private Assemblies	251
<i>Specifying Arbitrary Assembly Locations</i>	253

<i>Understanding the Shared Assembly</i>	254
Generating a Strongly Named Assembly	255
Prepping the Project Workspace	257
Building the Air Vehicles	258
Generating the Strong Name	260
<i>Using the Shared Assembly</i>	262
<i>Versioning Shared Assemblies</i>	264
Versioning the Shared VB .NET AirVehicles Assembly	265
<i>Working with Publisher Policy Assemblies</i>	267
The Binding Process in a Nutshell	269
<i>And Now for Something Completely Different: System.CodeDOM</i>	270
System.CodeDOM: Its Meaning in Life	272
<i>Introducing the System.CodeDOM Namespace</i>	272
The Types of System.CodeDOM	274
Building a CodeDOM Example	276
Building the Main() Function	277
Building the File via CreateFile()	280
Building the HelloClass (and Containing Namespace)	281
<i>Compiling the Assembly</i>	284
Running the Assembly (Using Late Binding)	285
Running Your Application	285
<i>Summary</i>	288
 <i>Chapter 6 .NET Types</i>	 289
<i>The Role of System.Object</i>	289
Overriding ToString()	290
The Two Faces of Equality	292
<i>Examining the .NET Data Type System</i>	294
System Data Type Language Mappings	296
<i>The Set of Custom .NET Types</i>	297
.NET Class Types	299
.NET Structure Types	300
.NET Interface Types	301
.NET Enumeration Types	301
<i>Building a Complex Code Library</i>	302
<i>Understanding Reflection</i>	304
<i>Working with System.Type</i>	304
Obtaining a Type Reference Using System.Object.GetType()	306
Obtaining a Type Reference Using the C# typeof Operator	306
Obtaining a Type Reference Using the Type Class	307
<i>The System.Reflection Namespace</i>	309

<i>Dynamically Loading an Assembly</i>	310
Enumerating Types in a Referenced Assembly	311
Enumerating Class Members	311
Enumerating Method Parameters	311
<i>Building a Custom .NET Type Viewer</i>	312
Showing Selected Type Details.....	315
Building the More Details Menu.....	316
<i>A Brief Word Regarding System.Reflection.Emit</i>	323
<i>Understanding .NET Attributes</i>	323
<i>Creating and Applying Custom Attributes</i>	325
Viewing the Applied Metadata.....	326
Restricting Attribute Usage	327
Assembly- (and Module-) Level Attributes	328
<i>Reading Attributes at Runtime</i>	330
Late Binding Under the .NET Platform	331
The Activator Class.....	331
Late Binding to a Private Assembly	332
Invoking Parameterized Methods.....	334
<i>Binding Late to Shared Assemblies</i>	335
Contrasting COM and .NET Late-Binding Syntax.....	336
<i>Summary</i>	338

Chapter 7 .NET-to-COM Interoperability– The Basics.....339

<i>A High-Level Overview of .NET-to-COM Interoperability</i>	339
Understanding the Role of the RCW.....	340
<i>Building an Interop Assembly–The Simplest Possible Example</i> ...	342
A Visual Basic .NET Client	346
<i>Converting Between COM IDL Data Types and Managed Data Types</i>	346
Working with Managed GUIDs (System.Guid)	348
Blittable and Non-Blittable Data Types.....	349
<i>Interfaces Consumed by the RCW</i>	351
<i>Options to Obtain an Interop Assembly</i>	353
Using the tlbimp.exe Command Line Utility	353
<i>Examining the Generated .NET Types</i>	356
Manipulating COM Types Using Generated “-Class” Types	357
Manipulating COM Types Using Discrete Interfaces	358
Manipulating COM Types Using the [Default] Interface Type	361

<i>Select Members of the</i>	
<i>System.Runtime.InteropServices Namespace</i>	362
An Interesting Aside:	
System.Runtime.InteropServices.RuntimeEnvironment	366
<i>COM Library Statement to .NET</i>	
<i>Assembly Statement Conversion Rules</i>	367
Programmatically Controlling the Namespace Definition	369
<i>COM Types to .NET Types Conversion Rules</i>	371
COM Interface Conversion.....	371
COM Interface Members to .NET Method Conversion Rules	375
COM Method Parameters to .NET Method	
Parameters Conversion Rules	377
Handling Optional and Default Parameters	381
COM Coclass Conversion	385
Mapping [noncreatable] and [appobject] Coclasses.....	387
Cataloging COM DISPIDs.....	388
Additional Coclass to .NET Class Infrastructure.....	389
COM Enum Conversion.....	391
<i>Deploying Interop Assemblies</i>	392
<i>Creating a Primary Interop Assembly</i>	393
<i>Reflecting on Interop Assembly Attributes</i>	396
Obtaining Type Information for a COM Wrapper Type.....	398
The Role of System.__ComObject.....	399
<i>Interacting with Well-Known COM Servers</i>	399
<i>Summary</i>	402

Chapter 8 .NET-to-COM Interoperability– Intermediate Topics.....403

<i>Handling the COM VARIANT</i>	403
<i>Building a VARIANT-Centric COM Server</i>	405
Exercising COM VARIANTs from Managed Code	407
The VARIANT Wrappers.....	409
<i>Handling COM SafeArrays</i>	410
Exercising COM SAFEARRAYs from Managed Code	413
Receiving SAFEARRAYs.....	416
<i>Handling C-Style Arrays</i>	419
<i>Handling COM Param Arrays</i>	420
<i>Handling COM Structures</i>	421
Building a VB 6.0 Structure Server	423
Exercising COM Structures from Managed Code	424
<i>Handling COM Collections</i>	426
Exercising the COM Collection from Managed Code	433

<i>A Brief Review of COM Connection Points (COM Events)</i>	437
Understanding IConnectionPointContainer	437
Understanding IConnectionPoint	438
Understanding the Outbound Interface	439
All Together Now...	440
<i>Building a Connectable COM Type</i>	441
<i>A Brief Review of .NET Delegates</i>	443
<i>A Brief Review of .NET Events</i>	445
<i>Examining the Interop Assembly</i>	448
Examining the Generated Delegates	450
Examining the Generated __CoCar and __CoCar_Event Interfaces	451
Examining the Managed CoClass	453
Receiving the COM Events (C#)	455
Receiving the COM Events (VB .NET)	456
Handling COM Types with Multiple [source] Interfaces.....	457
<i>Handling COM Error Objects</i>	459
Manipulating COM Error Objects in COM.....	461
The Role of ISupportErrorInfo	462
Building a Simple ATL Error Server	462
<i>The .NET Error Handling Mechanism</i>	464
The COMException Type.....	466
Handling COM Error Information from Managed Code	466
<i>Debugging COM Servers Using VS .NET</i>	468
<i>Summary</i>	470

Chapter 9 .NET-to-COM Interoperability– Advanced Topics.....471

<i>Revisiting the Marshal Class</i>	471
Directly Interacting with IUnknown.....	473
Manually Destroying a COM Object.....	474
<i>COM Coclasses Implementing .NET Interfaces</i>	475
Building a Comparable COM Class	476
Building a Cloneable COM Type.....	480
Building .NET Type Compatible Coclasses Using ATL 3.0	481
<i>Guidelines for Building .NET Type Compatible COM Classes</i>	484
Guideline #1: Don't Indirectly Implement the Same Interface Twice	485
Guideline #2: Don't Implement .NET Class Interfaces.....	485
Guideline #3: Take Care When Implementing Interface Hierarchies.....	487
Guideline #4: VB 6.0 Has a Key Limitation (Imagine That . . .).....	489

<i>Consuming ActiveX Controls from Managed Code</i>	490
Building an Example ActiveX Control	490
Viewing the Generated IDL	493
<i>Options for Consuming ActiveX Controls from Managed Code</i>	495
Consuming an ActiveX Control Using the VS .NET IDE	495
Importing ActiveX Controls Using AxImp.exe	501
<i>Modifying the Code for the AxHost-Derived Type</i>	504
One Final Modification	506
<i>Manually Modifying Interop Assemblies</i>	508
Building an Example COM Server	509
<i>Understanding the Interop Editing Process</i>	510
Generating the Initial Interop Assembly	511
Obtaining the *.il File for interop.HelpfulATLServer.dll	512
Viewing the Original IL/Metadata Definitions	514
<i>Dissecting the Layout of Attribute Metadata</i>	517
<i>Building a “Scratch” Assembly</i>	519
Creating a .NET [custom] Wrapper	519
Updating the Interop Assembly	522
Recompiling the IL	524
<i>Building the Managed Client</i>	526
<i>Building a Custom Type Library Importer Utility</i>	528
Understanding the TypeLibConverter Class	528
<i>Building the Main Shell</i>	530
Programmatically Loading the COM Type Information	531
Dissecting the TypeLibConverter.ConvertTypeLibToAssembly() Method	533
Building the Helper Sink	535
Implementing MyTlbImpApp.GenerateAssemblyFromTypeLib()	536
<i>Summary</i>	538

Chapter 10 COM-to-.NET Interoperability– The Basics

<i>The Role of the CCW</i>	539
The CCW: Mapping .NET Data Types into COM IDL Data Types	540
The CCW: Simulating Implemented COM Interfaces	542
The CCW: Simulating COM Identity	543
<i>Core Requirements for COM-to-.NET Communications</i>	544
Generating COM Type Definitions	545
Registering the .NET Assembly with COM	545
<i>Using the tlbexp.exe Utility</i>	546

<i>General Guidelines for Building COM-Aware .NET Types</i>	547
Establishing Type Visibility.....	547
Establishing Type Member Visibility	548
Controlling Type and Member Visibility Using the ComVisible Attribute..	548
Translating Constructors and C#-Style Destructors.....	549
Translating Abstract Base Classes	551
Translating Overridable Members.....	551
Inheriting Public Members	553
Translating Static and Constant Members.....	554
<i>Critical .NET-to-COM Conversion Details</i>	554
Converting Method Signatures	555
Converting .NET Properties	556
Converting .NET Fields.....	557
<i>Understanding the Class Interface</i>	557
Establishing a .NET Class Interface	559
The Custom IDL Attribute:	
{0F21F359-AB84-41E8-9A78-36D110E6D2F9}.....	561
<i>Understanding the Object Interface</i>	562
Transforming System.Object.ToString()	563
<i>The Case Against Class Interfaces</i>	563
<i>Exposing Custom .NET Interfaces to COM</i>	564
Controlling IID and DISPID Generation	566
<i>Implementing Explicit Interfaces</i>	567
<i>Controlling the Generated ProgID</i>	567
<i>Controlling the COM Library Definition</i>	568
<i>Handling Overloaded Methods</i>	569
<i>Importing mscorlib.tlb</i>	570
<i>Using the regasm.exe Utility</i>	572
<i>Examining the Updated Entries</i>	574
The ProgID Entry.....	575
The HKCR\CLSID Entry.....	575
Enlisting .NET Types into a Specific COM Category	577
Registering the Exposed Interfaces.....	579
Registering the COM Type Library.....	582
<i>Deploying the .NET Assembly</i>	582
<i>Leveraging the Visual Studio .NET IDE</i>	584
<i>Building a Visual Basic 6.0 COM Client</i>	584
Interacting with Mscorlib.tlb.....	586
<i>Building a C++ COM Client</i>	589
<i>Building a VBScript COM Client</i>	590
<i>Summary</i>	591

Chapter 11 COM-to-.NET Interoperability– Intermediate Topics	593
<i>Converting .NET Enums to COM Enums</i>	593
Altering the Underlying Field Storage	595
Leveraging System.Enum?	597
<i>Converting .NET Structures to COM Structures</i>	598
Exposing a .NET Structure As an IDL Union	603
<i>Converting .NET Delegates to COM Connection Points</i>	604
Creating the Event Interface	605
Specifying the Event Interfaces (a la ComSourceInterfacesAttribute)	606
Establishing Multiple [source] Interfaces	607
<i>Building a .NET Event Server Using VB .NET</i>	608
<i>Building a Visual Basic 6.0 Event Client</i>	609
<i>Building a C++ Event Client</i>	610
Building the Client-Side Sink	611
<i>Exposing Custom .NET Collections</i>	614
A VB 6.0 .NET Collection Client	617
<i>Exposing .NET Exceptions</i>	619
Throwing Custom .NET Exceptions	620
<i>Exercising Your DotNetCollection Assembly from C++</i>	621
<i>Converting .NET Interface with Multiple Base Interfaces</i>	624
<i>Converting .NET Interface Hierarchies</i>	627
<i>Summary</i>	630
 Chapter 12 COM-to-.NET Interoperability– Advanced Topics	 633
<i>Changing Type Marshaling Using MarshalAsAttribute</i>	633
<i>.NET Types Implementing COM Interfaces</i>	638
<i>Defining Custom COM Interfaces</i>	638
Building and Deploying the Interop Assembly	640
Building a Binary-Compatible C# Type	641
Building a Binary-Compatible VB .NET Type	642
Registering the .NET Assemblies with COM	644
<i>Building a VB 6.0 COM Client</i>	644
<i>Defining COM Interfaces Using Managed Code</i>	646
Selected Notes on Manually Defining COM Interfaces Using Managed Code	649
<i>Manually Defining COM Atoms: An Extended Example</i>	650
Defining the Dual Interface (and SHAPECOLOR Enum) Using C#	651

<i>Interacting with Interop Assembly Registration</i>	653
Inserting Custom Registration Information	654
<i>Programmatically Converting Assemblies</i>	
to COM Type Information.....	655
<i>Hosting the .NET Runtime from an Unmanaged Environment</i>	660
Building a Custom Host.....	663
<i>Summary</i>	667

Chapter 13 Building Serviced Components

(COM+ Interop)	669
<i>The MTS, COM+, Component Services Name Game</i>	669
<i>Recapping Component Services</i>	670
<i>Reviewing the COM+ Runtime Environment</i>	672
Reviewing Object Context	673
Reviewing Call Context	674
<i>The Role of the COM+ Catalog</i>	675
<i>The Component Service Explorer</i>	678
Creating COM+ Applications	679
<i>A Classic COM+ Example</i>	682
<i>Building a VB 6.0 COM+ Client</i>	683
<i>Deploying COM+ Applications</i>	685
<i>The System.EnterpriseServices Namespace</i>	687
The System.EnterpriseServices.ServicedComponent Type	689
<i>The Simplest Possible Example</i>	690
Installing the Code Library.....	693
<i>Using the regsvcs.exe Command Line Utility</i>	694
Viewing the Generated Type COM Information	695
Acknowledging the Registration Entries	696
And Finally, the COM+ Application Itself.....	697
<i>Accessing the Configured .NET Component from VB 6.0</i>	698
<i>Accessing the Configured .NET Component from C#</i>	699
<i>Enabling Component Statistics</i>	699
<i>A Brief Word on Lazy (Automatic) Registration</i>	700
<i>Working with the RegistrationHelper Type</i>	701
<i>Configuring a Managed COM+</i>	
Application Using .NET Attributes	703
<i>Supporting Object Construction Strings</i>	704
<i>Examining the ContextUtil Type</i>	706

<i>Understanding JITA</i>	708
The “Happy” and “Done” Bits	709
Enabling JITA/ASAP Deactivation	710
Controlling the Done Bit	712
JITA and Implementing IObjectControl (So to Speak...)	713
<i>JITA, IObjectControl, and the .NET Garbage Collector</i>	714
<i>Configuring Poolable Objects</i>	715
<i>A Recap of Transactional Programming</i>	717
Root Objects, Secondary Objects, and COM+ Transactions	718
<i>Programming COM+ Transactions</i>	720
Setting the Happy Bit	722
<i>A Complete Serviced Component Example</i>	724
Building the Custom Database	725
Building the C# Code Library	726
Building the LogSale Type	727
Building the CarInventory Class Type	728
Building the Windows Forms Front End	732
Building an ASP.NET Web Service Client	734
<i>Final Thoughts Regarding System.EnterpriseServices</i>	736
<i>Summary</i>	738
 <i>Index</i>	 739