COM and .NET Interoperability

ANDREW TROElsen
This book is dedicated to Mary and Wally Troelsen (aka Mom and Dad). Thanks for buying me my first computer (the classic Atari 400) so long ago and for staying awake during my last visit when I explained (in dreadful detail) how System.Object is so much better than IUnknown. I love you both.
Contents at a Glance

Acknowledgments ........................................................................................................... xxiii
Introduction ................................................................................................................... xxv

Chapter 1 Understanding Platform Invocation Services .......... 1
Chapter 2 The Anatomy of a COM Server .................................................. 51
Chapter 3 A Primer on COM Programming Frameworks ................. 127
Chapter 4 COM Type Information ................................................................. 161
Chapter 5 The Anatomy of a .NET Server .................................................. 229
Chapter 6 .NET Types ...................................................................................... 289
Chapter 7 .NET-to-COM Interoperability— The Basics ..................... 339
Chapter 8 .NET-to-COM Interoperability— Intermediate Topics ........ 403
Chapter 9 .NET-to-COM Interoperability— Advanced Topics ............ 471
Chapter 10 COM-to-.NET Interoperability— The Basics .................... 539
Chapter 11 COM-to-.NET Interoperability— Intermediate Topics .......... 593
Chapter 12 COM-to-.NET Interoperability— Advanced Topics ............. 633
Chapter 13 Building Serviced Components (COM+ Interop) .............. 669

Index ......................................................................................................................... 669
# Contents

Acknowledgments ........................................................................................................... xxiii
Introduction ....................................................................................................................... xxv

## Chapter 1  Understanding Platform Invocation Services

### The Two Faces of Unmanaged Code........................................................................... 1

### Understanding the C-Style DLL ................................................................................ 2

- Exporting Custom Members ......................................................................................... 3

### Building a Custom C-Based DLL ............................................................................... 5

- Functions Using Basic Data Types and Arrays ........................................................... 6
- Functions Receiving Structures (and Structures Containing Structures) .................... 7
- Functions Using Class Types ....................................................................................... 8

### Viewing Your Imports and Exports Using dumpbin.exe ........................................... 9

### Deploying Traditional DLLs ..................................................................................... 12

### A Dynamic C++ DLL Client .................................................................................... 15

- Dynamically Loading an External Library .................................................................... 15
- Invoking Members ........................................................................................................ 16

### The Atoms of PInvoke .............................................................................................. 18

- Data Type Conversions ............................................................................................... 18
- The Marshal Class ........................................................................................................ 20
- The DllImportAttribute Type ..................................................................................... 25

### A Trivial PInvoke Example....................................................................................... 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

### Interacting with MyCustomDLL.dll ....................................................................... 33

- Marshaling Arrays ...................................................................................................... 34
- Passing Structures ...................................................................................................... 35
- Receiving Allocated Structures .................................................................................. 37
- Interacting with Exported Class Types ...................................................................... 39
- Examining the Underlying IL ..................................................................................... 41

### Working with Unmanaged Callbacks ....................................................................... 42

### A Simple Callback Example.................................................................................... 43

### A More Interesting Callback Function ...................................................................... 44

### Building a C# Callback Client .................................................................................. 46

### Summary .................................................................................................................... 49
Chapter 2 The Anatomy of a COM Server ........................................51

Of Classes and Interfaces ................................................................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

The Composition of a COM DLL ..................................................63

The Role of Type Libraries ..........................................................65
  The Full MIDL Output ...............................................................66

The Role of the System Registry ..................................................66

Creating the COM DLL Project Workspace ..................................67

Understanding COM Interfaces ..................................................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

A Brief Word on COM-Centric Macros .........................................79

Implementing the ComCar ............................................................80
  Implementing IUnknown ..........................................................81
  Implementing ICar and IRadio ..................................................83

Understanding IClassFactory .....................................................84

Building Your Class Factory .......................................................85
  Implementing IClassFactory::CreateInstance() ..........................86
  Implementing IClassFactory::LockServer() ...............................87

Implementing DLL Component Housing ......................................88
  Managing Server Lifetime: DllCanUnloadNow() .........................88
  Contending with DllRegisterServer() and DllUnregisterServer() ....90

Exporting the Exports .................................................................90

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

Chapter 3 A Primer on COM Programming Frameworks ....................... 127
The Role of the Active Template Library .............................................. 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
# Contents

**The Role of Visual Basic 6.0** ................................................................. 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

**Summary** ............................................................................................. 159

### Chapter 4  COM Type Information ...................................................... 161

**The Format of a COM IDL File** .......................................................... 162

**Defining COM Types** ........................................................................... 163

**Primitive COM Data Types** ............................................................... 164
  - The Oieautomation (aka Variant) Data Types .................................... 166

**IDL Method Parameter Attributes** ..................................................... 167

**Defining COM Interface Types in IDL** ............................................... 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

**The ITypeInfo Interface** ...................................................................... 185
  - A Brief Word on ITypeInfo2 ............................................................. 188
  - Related ITypeInfo Data Types ......................................................... 188

**Generating COM Type Information Programmatically** ....................... 189
  - A Brief Word on ICreateTypeLib2 and ICreateTypeInfo2 .................. 190
  - The Target IDL ............................................................................... 190

**Building the Type Library (ICreateTypeLib)** ...................................... 191

**Creating the IHello Interface** ............................................................ 193
  - Breaking Down the CreateInterface() Helper Method .................... 195

**Building the SayHello() Method** ...................................................... 198

**Building the Hello Coclass** .............................................................. 200

**Testing the Application** .................................................................... 201
Programmatically Reading COM Type Information ...................... 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
Defining Custom IDL Attributes .................................................. 212
Reading Custom Attributes ....................................................... 214
Introducing the System.Runtime.InteropServices Namespace ...... 218
Building a C# COM Type Information Viewer ............................ 220
Loading the COM Type Library .................................................. 221
Loading the COM Type Library .................................................. 222
Displaying the COM Types ........................................................ 224
Summary .................................................................................. 228

Chapter 5 The Anatomy of a .NET Server ................................. 229
The Philosophy of .NET ............................................................. 230
The Building Blocks of .NET ..................................................... 231
Working with Managed Languages .......................................... 232
The Composition of a .NET Binary .......................................... 233
Building a C# Code Library ..................................................... 235
Building the Abstract Car Type .................................................. 238
Building the Derived MiniVan Type .......................................... 240
Implementing the Convertibles .................................................. 240
Establishing the Assembly Manifest ........................................... 242
Introducing ILDasm.exe ............................................................. 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
Building a Managed Client ....................................................... 250
Configuring Private Assemblies ............................................... 251
Specifying Arbitrary Assembly Locations ................................. 253
Dynamically Loading an Assembly .............................................................. 310
Enumerating Types in a Referenced Assembly .............................................. 311
Enumerating Class Members ......................................................................... 311
Enumerating Method Parameters ..................................................................... 311
Building a Custom .NET Type Viewer .......................................................... 312
Showing Selected Type Details ..................................................................... 315
Building the More Details Menu ................................................................... 316
A Brief Word Regarding System.Reflection.Emit ........................................... 323
Understanding .NET Attributes ..................................................................... 323
Creating and Applying Custom Attributes ..................................................... 325
Viewing the Applied Metadata ..................................................................... 326
Restricting Attribute Usage ......................................................................... 327
Assembly- (and Module-) Level Attributes ...................................................... 328
Reading Attributes at Runtime ..................................................................... 330
Late Binding Under the .NET Platform .......................................................... 331
The Activator Class ...................................................................................... 331
Late Binding to a Private Assembly ............................................................... 332
Invoking Parameterized Methods ................................................................... 334
Binding Late to Shared Assemblies ................................................................. 335
Contrasting COM and .NET Late-Binding Syntax .......................................... 336
Summary ......................................................................................................... 338

Chapter 7 .NET-to-COM Interoperability—The Basics ................................... 339
A High-Level Overview of .NET-to-COM Interoperability ................................ 339
Understanding the Role of the RCW ............................................................. 340
Building an Interop Assembly—The Simplest Possible Example .................... 342
A Visual Basic .NET Client ........................................................................... 346
Converting Between COM IDL Data Types and Managed Data Types ............ 346
Working with Managed GUIDs (System.Guid) ............................................... 348
Blittable and Non-Blittable Data Types ............................................................ 349
Interfaces Consumed by the RCW ................................................................. 351
Options to Obtain an Interop Assembly .......................................................... 353
Using the tlbimp.exe Command Line Utility .................................................. 353
Examining the Generated .NET Types ............................................................ 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
Select Members of the
System.Runtime.InteropServices Namespace .......................... 362
An Interesting Aside:
COM Library Statement to .NET
Assembly Statement Conversion Rules ............................. 367
Programmatically Controlling the Namespace Definition .... 369
COM Types to .NET Types Conversion Rules ....................... 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
Deploying Interop Assemblies ........................................ 392
Creating a Primary Interop Assembly .............................. 393
Reflecting on Interop Assembly Attributes ......................... 396
Obtaining Type Information for a COM Wrapper Type ......... 398
The Role of System.__ComObject .................................. 399
Interacting with Well-Known COM Servers ....................... 399
Summary ........................................................................ 402

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

Handling the COM VARIANT ........................................ 403
Building a VARIANT-Centric COM Server ....................... 405
Exercising COM VARIANTs from Managed Code ............... 407
The VARIANT Wrappers ............................................... 409
Handling COM SafeArrays .......................................... 410
Exercising COM SAFEARRAYs from Managed Code .......... 413
Receiving SAFEARRAYs .............................................. 416
Handling C-Style Arrays ............................................ 419
Handling COM Param Arrays ..................................... 420
Handling COM Structures .......................................... 421
Building a VB 6.0 Structure Server ................................. 423
Exercising COM Structures from Managed Code ............ 424
Handling COM Collections ......................................... 426
Exercising the COM Collection from Managed Code ....... 433
A Brief Review of COM Connection Points (COM Events) ..................437
  Understanding IConnectionPointContainer ..................................437
  Understanding IConnectionPoint ..................................................438
  Understanding the Outbound Interface .........................................439
  All Together Now...........................................................................440
Building a Connectable COM Type..................................................441
A Brief Review of .NET Delegates ..................................................443
A Brief Review of .NET Events .......................................................445
Examining the Interop Assembly .....................................................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 COMTypes with Multiple [source] Interfaces....................457
Handling COM Error Objects .......................................................459
  Manipulating COM Error Objects in COM .....................................461
  The Role of ISupportErrorInfo ...................................................462
  Building a Simple ATL Error Server ............................................462
The .NET Error Handling Mechanism ..............................................464
  The COMException Type ............................................................466
  Handling COM Error Information from Managed Code .................466
Debugging COM Servers Using VS .NET .........................................468
Summary .........................................................................................470

Chapter 9 .NET-to-COM Interoperability—Advanced Topics ..............471
Revisiting the Marshal Class ..........................................................471
  Directly Interacting with IUnknown ..............................................473
  Manually Destroying a COM Object .............................................474
COM Coclasses Implementing .NET Interfaces ..................................475
  Building a Comparable COM Class ..............................................476
  Building a Cloneable COM Type ..................................................480
  Building .NET Type Compatible Coclasses Using ATL 3.0 ..............481
Guidelines for Building .NET Type Compatible COM Classes ............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
Contents

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

Chapter 10 COM-to-.NET Interoperability—
The Basics .............................................................................. 539

The Role of the CCW ................................................................. 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
Core Requirements for COM-to-.NET Communications ............... 544
   Generating COM Type Definitions ........................................... 545
   Registering the .NET Assembly with COM .............................. 545
Using the tlbexp.exe Utility ........................................................ 546
General Guidelines for Building COM-Aware .NET Types ............... 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

Critical .NET-to-COM Conversion Details .................................. 554
    Converting Method Signatures .......................................... 555
    Converting .NET Properties ............................................ 556
    Converting .NET Fields .................................................. 557

Understanding the Class Interface ........................................... 557
    Establishing a .NET Class Interface ................................... 559
    The Custom IDL Attribute: {0F21F359-AB84-4lE8-9A78-36D110E6D2F9} ........ 561

Understanding the Object Interface ......................................... 562
    Transforming System.Object.ToString() .................................. 563

The Case Against Class Interfaces .......................................... 563

Exposing Custom .NET Interfaces to COM .................................... 564
    Controlling IID and DISPID Generation ................................ 566

Implementing Explicit Interfaces ............................................ 567

Controlling the Generated ProgID .......................................... 567

Controlling the COM Library Definition .................................... 568

Handling Overloaded Methods ................................................. 569

Importing mscorlib.tlb ...................................................... 570

Using the regasm.exe Utility ............................................... 572

Examining the Updated Entries .............................................. 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

Deploying the .NET Assembly .................................................. 582

Leveraging the Visual Studio .NET IDE .................................... 584

Building a Visual Basic 6.0 COM Client ................................... 584
    Interacting with Mscorlib.tlb .......................................... 586

Building a C++ COM Client ................................................... 589

Building a VBScript COM Client ............................................. 590

Summary ................................................................................. 591
Chapter 11 COM-to-.NET Interoperability—Intermediate Topics .............................................................. 593

Converting .NET Enums to COM Enums .............................................................. 593
Converting .NET Structures to COM Structures .................................................. 598
Converting .NET Delegates to COM Connection Points ....................................... 604
Building a .NET Event Server Using VB .NET .................................................. 608
Building a Visual Basic 6.0 Event Client ........................................................... 609
Building a C++ Event Client ............................................................................... 610
Exposing Custom .NET Collections ................................................................... 614
Exposing .NET Exceptions .................................................................................. 619
Exercising Your .NET Collection Assembly from C++ .................................... 621
Converting .NET Interface with Multiple Base Interfaces ............................... 624
Converting .NET Interface Hierarchies .............................................................. 627
Summary ........................................................................................................... 630

Chapter 12 COM-to-.NET Interoperability—Advanced Topics ........................ 633

Changing Type Marshaling Using MarshalAsAttribute ......................................... 633
Defining Custom COM Interfaces ........................................................................ 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
Building a VB 6.0 COM Client ........................................................................... 644
Defining COM Interfaces Using Managed Code ............................................... 646
Selected Notes on Manually Defining COM Interfaces Using Managed Code ....... 649
Manually Defining COM Atoms: An Extended Example .................................... 650
Defining the Dual Interface (and SHAPECOLOR Enum) Using C# ................. 651
Chapter 13 Building Serviced Components  
(COM+ Interop).................................................................669

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

Understanding JITA.............................................................................................................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

JITA, IObjectControl, and the .NET Garbage Collector ......................................................714

Configuring Poolable Objects .........................................................................................715

A Recap of Transactional Programming ...........................................................................717
   Root Objects, Secondary Objects, and COM+ Transactions ........................................718

Programming COM+ Transactions ..................................................................................720
   Setting the Happy Bit ....................................................................................................722

A Complete Serviced Component Example .......................................................................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

Final Thoughts Regarding System.EnterpriseServices .....................................................736

Summary ...........................................................................................................................738

Index ...................................................................................................................................739
Acknowledgments

As always, I must give a very real and heartfelt thanks to all of the fine people at Apress. First, thanks to Gary Cornell and Dan Appleman for building such a great place for writers to do their work. A mammoth thanks to Grace Wong for gently pushing me forward in order to get this book out on time and for putting up with me in general. And thanks to Stephanie Rodriguez and Hollie Fischer for their awesome work in spreading the word about Apress titles both at home and across the globe.

A huge thanks also goes to Ami Knox, Nicole LeClerc, Sofia Marchant, and Anne Friedman, all of whom did fantastic jobs smoothing over any grammatical glitches on my part. Thanks to Habib Heydarian and Eric Gunnerson for providing excellent technical assistance. Further thanks to Diana Van Winkle, Kurt Krames, and Tom Debolski for making the book look respectable and professional inside and out. Special thanks to Valerie Robbins for working on (yet another) tight deadline in order to index these chapters.

As for those individuals a bit closer to home, a heartfelt thanks to all my coworkers at Intertech, Inc. (http://www.intertech-inc.com), for making my “real job” a wonderful place to be. The previous praise does not apply to Tom Salonek, whom I still don’t care much for at all (... well, maybe just a little). Further thanks are in order for my family and friends for remaining patient when I became “just a bit grumpy” during the last month of this project. Last but not least, I must thank my wife Amanda for supporting me through yet another stint of sleepless nights and for remaining positive and encouraging when I was anything but. Thanks all!
Introduction

The funny thing about writing a book on COM and .NET interoperability is that one author could craft a five- to ten-page article describing the basic details that you must understand to get up and running with interop-related endeavors. At the same time, another author could write volumes of material on the exact same subject. So, you may be asking, how could this massive discrepancy between authors possibly exist?

Well, stop and think for a moment about the number of COM-aware programming languages and COM application frameworks that exist. Raw C++/IDL, ATL, MFC, VB 6.0, and Object Pascal (Delphi) each have their own syntactic tokens that hide the underbelly of COM from view in various ways. Thus, the first dilemma you face as an interop author is choosing which language to use to build the COM sample applications.

Next, ponder the number of .NET-aware programming languages that are either currently supported or under development. C#, VB .NET, COBOL .NET, APL .NET, PASCAL .NET, and so on, each have their own unique ways of exposing features of the CTS to the software engineer. Therefore, the next dilemma is choosing which language to use to build the .NET applications.

Even when you solve the first two dilemmas and choose the languages to use during the course of the book, the final dilemma has to do with the assumptions made regarding the readers themselves. Do they have a solid understanding of IDL and the COM type system? Do they have a solid understanding of the .NET platform, managed languages, and metadata? If not, how much time should be spend pounding out such details?

Given the insane combinations of language preferences and reader backgrounds, I have chosen to take a solid stance in the middle ground. If I have done my job correctly, you will walk away from this text with the skills you need to tackle any interop-centric challenge you may encounter. Also, I am almost certain you will learn various tantalizing tidbits regarding the COM and .NET type systems.

My ultimate goal in writing this book is to provide you with a solid foundation of COM and .NET interoperability. To achieve this goal, I have chosen to provide material that defines the finer details of the COM and .NET architectures. For example, over the course of the first six chapters, you will learn how to programmatically generate and parse COM IDL, dynamically generate C# and VB .NET source code on the fly (via System.CodeDOM), and build .NET applications that
can read COM type information. After all, when you need to build a software solution that makes use of two entirely unique programming paradigms, you had better have a solid understanding of each entity.

However, once this basic foundation has been laid, the bulk of this book describes the process of making COM and .NET binaries coexist in harmony. As an added bonus, I cover the process of building .NET code libraries that can leverage the services provided by the COM+ runtime layer (via System.EnterpriseServices).

Now that you have the big picture in your mind, here is a chapter-by-chapter breakdown of the material:

**Chapter 1: Understanding Platform Invocation Services**
I open this examination of COM/.NET interoperability by focusing on the role of a single .NET class type: DllImportAttribute. In this chapter, you learn how to access custom C-based (non-COM) DLLs as well as the Win32 API from a managed environment. Along the way, you investigate how to marshal C structures, interact with traditional callback functions, and extract exported C++ class types from within a managed environment. This chapter also examines the role of the Marshal class, which is used in various places throughout the book.

**Chapter 2: The Anatomy of a COM Server**
The point of this chapter is to document the internal composition of a classic COM server using raw C++ and COM IDL. Given that many COM frameworks (such as VB 6.0) hide the exact underpinnings of COM, this chapter also examines the use of the system registry, required DLL exports, the role of the class factory, late binding using IDispatch, and so on. As you might guess, the COM server you construct during this chapter is accessed by managed code later in the text.

**Chapter 3: A Primer on COM Programming Frameworks**
Given that you build a number of COM servers over the course of the book, this (brief) chapter provides an overview of two very popular COM frameworks: the Active Template Library (ATL) and Visual Basic 6.0. Knowledge mappings are made between the raw C++ server created in Chapter 2 and the binaries produced by the ATL/VB 6.0 COM frameworks. Along the way, you also explore the key COM development tool, oleview.exe.

**Chapter 4: COM Type Information**
This chapter examines the gory details of the COM type system, including a number of very useful (but not well-known) tasks such as constructing custom IDL attributes, applying various IDL keywords such as [appobject], [noncreatable], and so forth. More important, this chapter also illustrates how to read and write COM type information programmatically using ICreateTypeLibrary, ICreateTypeInfo,
and related COM interfaces. This chapter wraps up by examining how to build a managed C# application that can read COM type information using interop primitives.

**Chapter 5: The Anatomy of a .NET Server**
The goals of this chapter are to examine the core aspect of a .NET code library, including various deployment-related issues (for example, XML configuration files, publisher policy, and the like). This chapter also provides a solid overview of a seemingly unrelated topic: dynamically generating and compiling code using System.CodeDOM. Using this namespace, developers are able to dynamically generate code in memory and save it to a file (*.cs or *.vb) on the fly. Once you have investigated the role of System.CodeDOM, you will have a deeper understanding of how various interop-centric tools (such as aximp.exe) are able to emit source code via command line flags.

**Chapter 6: .NET Types**
If you haven’t heard by now, understand that the .NET type system is 100 percent different than that of classic COM. Here, you solidify your understanding of the .NET type system, including the use of custom .NET attributes. This chapter also examines the role of the System.Reflection namespace, which enables you to dynamically load an assembly and read the contained metadata at runtime. This chapter also illustrates late binding under .NET and the construction of custom managed attributes. I wrap up by showing you how to build a Windows Forms application that mimics the functionality provided by ILDasm.exe.

**Chapter 7: .NET-to-COM Interoperability—The Basics**
In this chapter, the focus is on learning how to build .NET applications that consume classic COM servers using a Runtime Callable Wrapper (RCW). You begin with the obvious (and most straightforward) approach of using the integrated wizards of Visual Studio .NET. Next, you learn about the tlbimp.exe tool (and the numerous command line options). Along the way, you are exposed to the core conversion topics, including COM/.NET data type conversions, property and method mappings, and other critical topics.

**Chapter 8: .NET-to-COM Interoperability—Intermediate Topics**
This chapter builds on the previous one by examining a number of intermediate topics. For example, you learn how .NET clients can make use of COM VARIANTs and SafeArrays, COM Error Objects, COM enums, COM connection points, and COM collections. Topics such as exposing COM interface hierarchies are also examined in detail.
Chapter 9: .NET-to-COM Interoperability—Advanced Topics
Here you learn to import ActiveX controls and augment the work performed by the aximp.exe command line utility to account for COM [helpstring] attributes that are lost during the conversion process. Furthermore, this chapter examines the process of manually editing the metadata contained in a given interop assembly. For example, you learn how to support [custom] IDL attributes in terms of .NET metadata and understand how to compile *.il files using ilasm.exe. This chapter also describes how a COM type can implement .NET interfaces to achieve “type compatibility” with other like-minded .NET types. You wrap up by learning how to build a custom type library importer application using C#.

Chapter 10: COM-to-.NET Interoperability—The Basics
This chapter focuses on how COM clients (written in VB 6.0, C++, and VBScript) can make use of .NET types using a COM Callable Wrapper (CCW). Here, I cover class interfaces, the tlbexp.exe/regasm.exe command line tools, and various registration and deployment issues. This chapter also examines how a COM client can interact with the types contained in the core .NET assembly, mscorlib.dll.

Chapter 11: COM-to-.NET Interoperability—Intermediate Topics
This chapter builds on the materials presented in Chapter 10 by examining how .NET enumerations, interface hierarchies, delegates, and collections are expressed in terms of classic COM. You also learn how to expose custom .NET exceptions as COM error objects, as well as about the process of exposing .NET interface hierarchies to classic COM.

Chapter 12: COM-to-.NET Interoperability—Advanced Topics
This advanced COM-to-.NET-centric chapter examines how a .NET programmer is able to build “binary-compatible” .NET types that integrate with classic COM. You see how a .NET type can implement COM interfaces, and you also get a chance to explore the details of manually defining COM types using managed code. This chapter also examines how to interact with the registration process of an interop assembly. The final topics of this chapter address the process of building a custom host for the .NET runtime (using classic COM) and the construction of a custom .NET-to-COM conversion utility.

Chapter 13: Building Serviced Components (COM+ Interop)
Despite the confusion, .NET programmers are able to build code libraries that can be installed under COM+. In this final chapter, I begin by examining the role of the COM+ runtime and reviewing how it fits into n-tier applications. The bulk of this chapter is spent understanding the System.EnterpriseServices namespace and numerous types of interest. You learn how to program for JITA, object pools,
construction strings, and transactional support using managed code. I wrap up by constructing an n-tier application using managed code, serviced components, Windows Forms, and ASP .NET.

Now that you have a better understanding about the scope of this book and the mindset I have regarding the material that follows, understand that I have written this book based on the following assumptions about you:

- You are not satisfied with clicking a button of a given wizard and thinking “I guess it worked . . . somehow . . . I think.” Rather, I assume you would love to know the inner details of what that wizard does on your behalf and then click the button.

- You are aware of the role of COM, have created a number of COM servers, and feel confident building COM solutions in the language mapping of your choice. As well, I am assuming that you still find the process of learning the finer details of COM a worthwhile endeavor. As you will see, most of the COM servers built during the course of this book make use of VB 6.0, unless a particular COM atom cannot be expressed using the vernacular of BASIC. In these cases, I make use of the ATL framework.

- You are aware of the role of .NET, have (at the very least) explored the syntax of your favorite managed language, and (at the very most) created a number of .NET applications during the process. While many of my managed examples make use of C#, I also make use of VB .NET when necessary.

Finally, be aware that the source code for each example can be obtained from the Apress Web site in the Downloads section at http://www.apress.com.

It is my sincere hope that as you read though the text you enjoy yourself and expand your understanding of COM, the .NET platform, and the techniques used to blend each architecture into a unified whole.

Andrew Troelsen
Minneapolis, Minnesota