ADO.NET and ADO Examples and Best Practices for VB Programmers, Second Edition

WILLIAM R. VAUGHN

APress Media, LLC
This book is dedicated to those people in uniform who risk more than their careers every time they go to work. We are all indebted to them.

And to Charles Vincent Heintzelman: Veteran, Father, Grandfather, and Great Grandfather. He will be missed.
Foreword

Bill Vaughn has long been one of our most valuable contributors at Fawcette Technical Publications, Inc. He's written feature articles, columns, and opinion pieces for Visual Basic Programmer's Journal and later for Visual Studio Magazine. Bill has also been one of our most popular speakers at our VSLive! and VBITS conferences from Tokyo to Munich.

I remember Bill mainly for his bad sense of humor. During a presentation in Tokyo, where we had a team of skilled translators set to handle the challenging job of doing live translation of his technical presentation from English to Japanese, he started his presentation in Spanish.

But our readers and conference attendees remember Bill—and keep demanding his articles and presentations—for other reasons. Bill is one of the recognized authorities on data access using Microsoft's tools. Having encyclopedic knowledge of a topic as deep as data access is important, but what makes Bill's content, and no doubt this book, so valuable is his years of practical experience. Bill's writings get beyond the mere feature lists that fill so many computer books, to provide practical insight on using these tools in the real world.

Bill is also opinionated and candid. He sticks his neck out to provide honest opinions, rather than taking the safe route and toeing the party line. While you or I may not always agree with him, his candor is invaluable in helping you to avoid and understand potential pitfalls in your development. From the nitty-gritty of Recordsets and cursors, to providing insights on choosing an architectural approach, Bill is one of the most respected authors covering data access.

Enjoy this book. But don't laugh at his jokes—it just encourages him.

Jim Fawcette
President
Fawcette Technical Publications, Inc.
http://www.fawcette.com
# Contents at a Glance

**Foreword** ...................................................................................................................................... iv

**About the Author** ....................................................................................................................... xiii

**Acknowledgments** ..................................................................................................................... xv

**Introduction** ................................................................................................................................ xvii

**Part One**  **ADO Classic**.......................................................................................................... 1

  Chapter 1  *Working with ADO* .................................................................................................. 3

  Chapter 2  *Introduction to ADO* ............................................................................................... 15

  Chapter 3  *Creating ADO Objects* ............................................................................................. 23

  Chapter 4  *Getting Connected* .................................................................................................. 29

  Chapter 5  *ADO Command Strategies* ......................................................................................... 81

  Chapter 6  *Recordset Strategies* ................................................................................................. 139

  Chapter 7  *Manipulating Your Recordset* ................................................................................... 225

  Chapter 8  *Understanding Record and Stream Objects* ............................................................... 293

  Chapter 9  *Passing Resultsets Between Layers* ......................................................................... 301

  Chapter 10  *Getting Your Data into Shape* ............................................................................... 345

  Chapter 11  *Data Access Tips and Techniques* ......................................................................... 361

  Chapter 12  *ADO and the Visual Database Tools* ................................................................. 379

**Part Two**  **ADO.NET and the .NET Framework** ..................................................................... 399

  Chapter 13  *Introducing ADO.NET* ............................................................................................ 401

  Chapter 14  *ADO.NET—Getting Connected* .............................................................................. 435

  Chapter 15  *ADO.NET Command Strategies* ............................................................................. 473

  Chapter 16  *ADO.NET DataReader Strategies* ......................................................................... 529

  Chapter 17  *Using the DataTable and DataSet* ........................................................................ 545

  Chapter 18  *Filtering, Sorting, and Finding* ............................................................................. 585
| Chapter 19 | ADO.NET Update Strategies | 607 |
| Chapter 20 | ADO.NET Constraint Strategies | 667 |
| Chapter 21 | ADO.NET Error Management Strategies | 685 |
| Chapter 22 | ADO.NET and XML | 697 |
| Appendix A | Working with the Biblio Sample Database | 729 |
| Appendix B | CompareCursorLocation Chart | 731 |
| Appendix C | ADOc Dynamic Properties | 743 |
| Index | | 753 |
Contents

Foreword ................................................................................................................ iv

About the Author ........................................................................................... xiii

Acknowledgments ............................................................................................ xv

Introduction .................................................................................................... xvii

Part One ADO Classic ................................................................................. 1

Chapter 1 Working with ADO ................................................................. 3

Choosing the “Right” Version of ADO ....................................................... 3
What’s New in ADO 2.5? .............................................................................. 6
What’s New in ADO 2.6? .............................................................................. 8
What’s New in ADO 2.7? .............................................................................. 9
The ADO Component Checker .............................................................. 9
Installing ADO ......................................................................................... 10
Deploying ADO ....................................................................................... 11
The .NET Framework Approach .......................................................... 13
Getting More Help with ADO ............................................................... 14

Chapter 2 Introduction to ADO ............................................................... 15

ADO and OLE DB ..................................................................................... 15
The ADO Object Model ....................................................................... 18

Chapter 3 Creating ADO Objects ....................................................... 23

Instantiating ADO Objects ....................................................................... 23
Instantiating ADO Components in Visual Basic Script ....................... 25
When Things Go Wrong ....................................................................... 27
Chapter 4  Getting Connected ..........................................................29

The ADO Connection Object ...............................................................................31
Using the Connection Object ..........................................................................34
Managing Transactions ......................................................................................74

Chapter 5  ADO Command Strategies .....................................................81

Inner Workings of the Command Object .........................................................82
Building Command Objects .............................................................................86
Managing Command Objects ...........................................................................124
Performance Considerations: Using Command Objects Wisely ........137

Chapter 6  Recordset Strategies .................................................................139

Introducing Recordsets .....................................................................................139
Constructing Recordset Objects ....................................................................144
Working with Recordset Objects ....................................................................148

Chapter 7  Manipulating Your Recordset ..................................................225

Data Binding Your Recordset .........................................................................225
Understanding the State Property ..................................................................226
Fetching Techniques .........................................................................................227
Sorting, Filtering, and Finding .........................................................................243
Disconnecting Recordsets ...............................................................................248
Choosing Read-Only vs. Read-Write Recordsets ............................................249
Working with Field Objects .............................................................................249
Working with Stored Procedures ...................................................................261
Accessing Deprecated Recordsets ...................................................................287
Recordset Events ..............................................................................................289

Chapter 8  Understanding Record and Stream Objects 293

Introducing the ADO Record Object .................................................................293
Introducing the Stream Object .........................................................................294
Chapter 9 Passing Resultsets Between Layers .......... 301
Delivering Pizza with a Dump Truck ........................................ 302
Understanding Recordset Marshaling .................................. 306
Passing Disconnected Recordsets ......................................... 312
Passing Strings and String Arrays ........................................ 318
Passing OUTPUT Parameters Instead of Rowsets ................... 321
Passing Variant Arrays .......................................................... 324
Passing User-defined Structures ........................................... 326
Passing PropertyBags ............................................................. 338
Passing XML ............................................................................. 341

Chapter 10 Getting Your Data into Shape .................. 345
The Shape Provider .................................................................... 346
Building a Simple Shape ....................................................... 347
Why Not Use the Data Environment Designer? ..................... 353
Analyzing the Impact ............................................................... 356
Finding a Better Way ............................................................... 357
The Impact of the “Just Enough” Approach .............................. 359

Chapter 11 Data Access Tips and Techniques ............. 361
Referencing Data on Other Servers ........................................ 361
TSQL Tips ............................................................................... 362
Performance-Tuning Stored Procedures ............................... 365
SET NOCOUNT Using OLE DB Providers ................................. 366
Granting Permissions ................................................................. 367
More Recordset Secrets .......................................................... 369
Working with the Grid Control ............................................... 374
Working with Strings ............................................................... 375
Working with Graphics ............................................................. 377
MDAC/ADO Setup Issues ......................................................... 377
Hooking up Visual Basic/ADO Help ....................................... 378

Chapter 12 ADO and the Visual Database Tools .......... 379
Using the Data View Window .................................................. 380
Using the Data Environment Designer ................................. 386
Using ADO without the Data Environment Designer ............. 394
Using the Data Object Wizard ............................................... 395
Part Two ADO.NET and the .NET Framework

Chapter 13 Introducing ADO.NET .................................................. 401
How We Got Here ........................................................................... 401
ADO.NET—A New Beginning .......................................................... 405
Understanding ADO.NET Infrastructure ....................................... 407
ADO.NET .NET Data Providers ..................................................... 413
ADO.NET andDisconnected Data Structures .................................. 418
The System.Data Namespace ....................................................... 419
ADO.NET’s Low-Level Data Stream ............................................. 432

Chapter 14 ADO.NET—Getting Connected ................................ 435
ADO.NET .NET Data Providers ..................................................... 436
Creating ADO.NET Connection Objects ....................................... 438
Connecting via the SqlClient .NET Data Provider ......................... 446
Managing Transactions .................................................................. 453
Connecting with the Odbc .NET Data Provider ............................ 454
Opening ADO.NET Connections in Code ...................................... 457
Using the IDE to Get Connected .................................................. 458
Closing Connections ..................................................................... 460
COM-based ADO as a .NET Data Provider .................................. 468

Chapter 15 ADO.NET Command Strategies .............................. 473
Understanding ADO.NET Commands ........................................... 473
Binding to Resultset Columns ..................................................... 524

Chapter 16 ADO.NET DataReader Strategies ............................. 529
Comparing ADOc with ADO.NET .................................................. 530
Understanding the DataReader ................................................... 532
Using the DataReader Get Methods to Retrieve Data ................. 539
Cleaning Up After your DataReader ............................................ 543
Chapter 17 Using the DataTable and DataSet .......... 545

Understanding the DataSet and DataTable Structure .......... 547
Importing ADOc Recordsets .............................................. 555
Building DataTables ....................................................... 559
Accessing DataTable Data ............................................... 567
Accessing DataSet Data ................................................... 581
Binding to DataSets .......................................................... 582

Chapter 18 Filtering, Sorting, and Finding ............. 585

Filtering and Sorting with the Select Method .......... 586
Working with the DataViewManager ......................... 598
Using the Find Methods ............................................... 600

Chapter 19 ADO.NET Update Strategies .................. 607

Changing Data—The Fundamentals .................................. 608
Update Strategies .......................................................... 622
Using Visual Studio to Generate Action Queries .......... 633
Retrieving Autonumber, Identity, or GUIDs ............... 641
Understanding the Update Methods .............................. 645
Update Performance ....................................................... 647
Updating the Authors Table—An Example .................. 649
Update Error Handling Strategies ................................. 656
Handling Update Method Exceptions ......................... 660
Executing Ad Hoc Queries to Perform Updates ........... 664
Merging and Transporting Data ....................................... 664

Chapter 20 ADO.NET Constraint Strategies .......... 667

Understanding .NET Constraints .................................... 667
How ADO.NET Implements Constraints .................. 669
Creating DataRelation Objects ....................................... 679

Chapter 21 ADO.NET Error Management Strategies .... 685

Structured Exception Handling ...................................... 686
Understanding Try/Catch .................................................. 686
Trapping Specific vs. General Exceptions ................ 690
The System.Data Exceptions ........................................... 690
Displaying the Error Message ......................................... 695
Chapter 22 ADO.NET and XML .........................................................697

XML Support in ADO.NET ................................................................698
Standardizing on XML Formats ......................................................699
Understanding XML and DataSet Objects ....................................700
Creating a Web Service as an XML Data Source .........................710
Passing an Updated DataSet Back to a Web Service .....................720
Creating Custom Error Handlers ..................................................725
ADO.NET and SOAP ......................................................................726

Appendix A Working with the
Biblio Sample Database .............................................................729

Appendix B CompareCursorLocation Chart ...............................731

Appendix C ADOc Dynamic Properties ....................................743

Index ..............................................................................................753
About the Author

I'VE BEEN WORKING IN the computer industry since 1972. My first job was teaching FORTRAN and COBOL labs at Central Texas College in Killeen, Texas. After graduating, I worked in the mainframe industry for about seven years—desperately trying to stay sane. During this time I designed and implemented a number of DBMS systems, some of which might still be in use today. When I built my first microcomputer (while working for EDS in Dallas), Ross Perot found out and ended up hiring me to work for him in the Business Systems Division. I worked in the BSD for several years in the very early days of the PC industry. When this division folded, I surfed the industry for many years, working for companies large and small. These included Mostek (integrated circuits and systems manufacturer), Challenge Systems (a PC startup), Digital Research, and finally CPT Corporation. All of these companies were successful for awhile, and then succumbed to the heavy waves of competition and innovation. I worked in systems design, architecture and integration, marketing, training, and customer support. I supported, designed, and coded systems software, applications, and operating systems. I also managed two- to twenty-person teams of developers, mentored hundreds, but also spent quite a bit of time answering the phone to support PC novices and experts alike.

In 1986 I joined Microsoft and started by supporting Windows 1.0 developers. About a year later I started writing courses, training, and managing trainers at Microsoft University. There I met and mentored developers from all over the industry and learned a great deal about DBMS implementations. After Microsoft University folded, I wrote my first book on Visual Basic and data access, *The Hitchhiker's Guide to VBSQL*, and published it myself. This first book helped me get my next job on the Visual Basic documentation team at Microsoft. I wrote data access content for them for five years and wrote five more editions of my book. Before long I was “promoted” to work in the marketing group. After a couple of years of speaking, writing, and marketing Visual Basic 5.0 and 6.0, I moved to the MSTE internal training group where I worked for two years before my retirement in August of 2000. Since then I have earned an honorary Ph.D. from the University of Advancing Computer Technology in Tempe, Arizona.

I married in 1968 while at flight school—just before I went to Vietnam to fly helicopters. My wife Marilyn earned her bachelor's degree in mathematics and taught for many years before she got her master's degree in math and environmental studies. She is currently the treasurer of Beta V Corporation and does a great job managing the business side of my book and training business.

My younger daughter, Christina (a.k.a. Fred), graduated cum laude from Whitman College with an English and classics degree. She is currently working as
an editor for Beta V Corporation; she copy edited this book. Her interests include indoor and outdoor soccer; she's playing on and coaching several teams.

My older daughter, Victoria (a.k.a. George), graduated from the University of Washington with a master's degree in environmental engineering. I had to send her back to the Civil Engineering school at the University of Washington because her chemical engineering degree left her far from “civil.” Last March, Victoria and her husband, 1st Lt. Michael Ballard, gave birth to my first granddaughter Mary Anne Ballard.

My other interests include digital photography and travel. I’m currently developing a CD of Windows wallpaper pictures that illustrates the best of the places my lectures have taken me.
Acknowledgments

This book has taken longer to create than any of my previous books, due in part to the fundamental differences between Visual Basic 6.0 and its version of ADO and the new .NET paradigm. Throughout this process, I asked for and received lots of help. While some contributed more than others, all were generous with their time, talents, guidance, patience, and inside information. Microsoft was very helpful in many ways and should be proud of their Visual Studio .NET and SQL Server 2000 teams as represented by (in alphabetical order) Andrew Jenks, Ann Morris, Ari Bixhorn, Brad Merrill, Brian Henry, Chris Dias, Dave Mendlen, David Sceppa, David Schleifer, Drew Fletcher, Ed Robinson, Edward Jezierski, Jonathan Hawkins, Michael Pizzo, Mike Iem, Omri Gazitt, Paul Vick, Richard Waymire, Rick Nasci, Rob Copeland, Rob Howard, Robert Green, Sean Draine, Sidney Higa, and Stacey Giard. Microsoft provided training sessions, beta software, and code reviews that made the process of getting accurate information (not always favorable to Microsoft) in the book. This is a departure from Microsoft's prior approach to new versions of Visual Studio—and we authors really appreciate it. Others outside of Microsoft were also helpful—most of all Kimberly Tripp Simonnet and the Aces team, including Charles Carroll.

I would also like to mention the cadre of technical editors I enlisted to take on the challenges of learning an entirely new language, learning a basically new development suite, working with pre-beta and beta code, and doing so at great distances. Early on, David Jezak, an old friend and co-worker from the Visual Basic development team at Microsoft, took on the monumental task of building the example CD. He was able to reorganize it and repeatedly test the examples on each version of .NET as they emerged. Carl Prothman and Pamela Fanstill also provided repeated in-depth technical reviews of the .NET code and the prose. As the project was nearing completion, Peter Blackburn made another pass at the prose and the examples as he converted the Visual Basic edition to work in C#. All of these folks added significant value to the book because they made sure the words were right and that the examples did what they were intended to do. Thanks to all of you.

Since writing the first edition of ADO Examples and Best Practices, I have set up my own company to manage the business side of being an author and trainer. My wife, Marilyn, has been instrumental in keeping the company books and business in the black and on the right side of the (rather complex) tax laws. She has also been able to let me live and work in the same house with her, which is quite an accomplishment in and of itself given my grumpy nature when I’m working—but if she comes in here again with the vacuum . . .
This is the eighth book I’ve written and each time I’ve had to work with publisher-provided copy editors. Sometimes these folks were very good and sometimes . . . well, not as good. For this book, I was fortunate enough to be able to hire my own daughter, Christina (Fred), to do the copy edit. Having her work with me on a daily basis has been a challenge for both of us, but a prideful joy for me. I think she did very well, but I’ll let you, the reader, be the judge of her skill and dedication. Throughout the project, Chrissy also kept me on schedule and working on the correct files (no small task). More importantly, she made sure the words made sense and the jokes were funny (at least the ones she understood). Without her, I don’t know that this book would have been written on time—if at all. Thanks, Fred. It’s been a kick.
Introduction

For those of you new to my earlier book, ADO Examples and Best Practices, it was originally written as an update to my popular Hitchhiker's Guide to Visual Basic and SQL Server—6th Edition. Happily, the first edition of ADO Examples and Best Practices received glowing praise from a variety of reviewers, trainers, and developers all over the world. The first edition was not designed as a tutorial on basic ADO; it assumed you had at least some experience with ADO (there are lots of elementary ADO books). In contrast, this new edition adds quite a bit more introductory material to fill in some of the spaces not covered by the introductory books. It also leads the way for developers contemplating the process of converting existing COM-based ADO code to ADO.NET.

ADO.NET and ADO Examples and Best Practices for VB Programmers, Second Edition, focuses on ways to make your applications more efficient and at the same time help you write more efficient code in less time. These efficiencies can make the difference between a successful application (or component or Web page) and having to spend your weekends fixing its problems.

It's been several years since the first edition was published and what Microsoft was going to call "Visual Basic 7.0" is still not released to manufacturing—and it's probably never going to be. Instead, Microsoft has decided to forego any further changes to Visual Basic 6.0 in favor of creating an entirely new product—Visual Studio .NET. While I expect that some of you are looking forward to the innovations introduced in the Visual Studio .NET tools and languages, I was not so anxious to start learning a new language. I'm like the old carpenter who's used to his tried-and-true hammer. When a new-fangled tool comes to the local hardware store, I'll be one of the first to try it out, but one of the last to trust it. I've had too many hair-pulling, gut-wrenching, marriage-threatening experiences with version 1 tools to depend on them.

Yes, .NET eliminates a whole set of serious issues that have "bugged" Visual Basic developers. I expect, however, that we'll see an entirely new and yet-to-be-discovered set of issues that will make our work even more challenging—at least until version 3. I've detailed a bevy of these in the ADO.NET section of this book (beginning with Chapter 13). I also expect that ASP (Visual Basic Script) developers will be far more inclined to jump on the .NET bandwagon because it directly addresses many more of their language and architecture issues. But, to tell the truth, they never really programmed in "Visual Basic" in the first place.

About six months ago, I suggested to Apress that we clone this book and rework it for C# programmers. They thought this was a great idea and got right on it. Before long, Apress asked Peter Blackburn, a dynamite developer and writer
from the U.K., to do the conversion. He's done a great job of adapting my Visual Basic point of view to C#. This work is published in the *ADO.NET Examples and Best Practices for C# Programmers*.

As the title suggests, *ADO.NET and ADO Examples and Best Practices for VB Programmers*, Second Edition, includes an entirely new section devoted to ADO.NET. Unlike earlier sections on COM-based ADO, the ADO.NET section starts out at a more elementary level because I can't assume many developers out there are that familiar with this new technology. I've written this second half of the book with both experienced and novice ADO developers in mind. Throughout these chapters you'll find comparisons to COM-based ADO that can help make your transition to Visual Basic .NET and ADO.NET less daunting.

I see this conversion process from two points of view. First, you, your co-workers, and your managers have to learn new ways of creating solutions using basically new architectures. No, your business problems haven't really changed that much, but you'll find that the .NET Framework provides a number of new approaches to addressing these problems. The next phase of the conversion process is to consider what parts of your existing code can be brought forward into your new .NET applications. I expect that there are many applications, components, controls, forms and reports that won't be easy (or even possible) to convert. On the other hand, .NET does know how to access COM components through its COM interop technology. How well this works and what new issues this raises is an underlying thread in many of the ADO.NET chapters (13–22). Frankly, I would not rely on the conversion process working particularly well at first. Keep in mind that COM DLL hell is, well, hell.

Over the past year, I've come up with and learned a number of new "best practices" when working with .NET. Unfortunately, some of these (but not nearly all) contradict best practices espoused in the first section of the book. To make things easy, I begin with a focus on transitions from COM-based ADO and design alternatives that can't help but improve performance and scalability. I've also gleaned a number of excellent suggestions for good designs and the answers to typical questions based on my own testing, literally thousands of e-mail threads, and a number of .NET conference sessions. Microsoft was more than helpful as well; without their support, this book would never have been published and I would have taken up a new career by now.

COM-based ADO or ADO "Classic" (later in this book I'll refer to this version as ADOc) has also matured a great deal since I wrote the first edition of this book. I've made an in-depth pass at the first edition and I have updated, corrected, and tuned the text and code to better reflect how the latest version of ADO (2.7) works. There are several new features in ADO that make handling XML easier—not to mention the copious bug fixes. ADO 2.7 seems to be the same basic object model as 2.6, but with built-in .NET support. There is a rumor that it also has native 64-bit support.
Some of you will be migrating from other programming interfaces, such as Data Access Objects (DAO) or Remote Data Objects (RDO). On occasion, I make a passing comment about how to make this transition easier, but I don’t dwell on these issues. Some of you will be transitioning (whether you want to or not) from earlier versions of ADO—this book’s for you.

Obviously, *ADO.NET and ADO Examples and Best Practices for VB Programmers*, Second Edition, focuses on ActiveX Data Objects (ADO) version 2.7, but in a way that reflects what countless developers, trainers, and support engineers have passed on to me over the years. Yes, it’s these people that provide the pathway to what I write in these books. I have made a living over the last fifteen years by writing, training, and documenting code—not by writing or supporting applications (as I did the first twelve years of my career). Because of this, I must constantly work with others who do. When I can, I help them solve problems encountered in the “real world,” but just as often, the developer community provides me with innovative and tried-and-true ways of using the tools and languages. These examples and “best practices” are techniques discussed, refined, and used by hundreds (or thousands) of the most talented developers in the world. And yes, many of the ideas come from “international” developers outside of the U.S. While some of the ideas come from inside Microsoft—where I used to be bound to a certain level of confidentiality—many come from off-campus, where developers often have to make do with what the folks on-campus produce and shove out (and under) the door. I also communicate on a regular basis with the product managers, developers, and product support teams at Microsoft, trying to get the tools and interfaces to work better—based on your requests (and mine).

This edition assumes that most Web developers will want to migrate to .NET, so it does not spend a lot of time dwelling on ASP development. While the book covers the new ADO 2.7 features, it also discusses other technologies such as XML and MSXML whenever it makes sense to do so. While this book is not a “programmer’s reference,” it does detail many of the individual objects, properties, and methods similar to my strategy in writing the *Guide*. Despite the Web hype out there, this book fully discusses client/server and middle tier, along with Web-based, ADO architectures, and the code development, deployment, and performance analysis issues associated with each paradigm. Why do I keep focusing on client/server? Well, at each conference I attend, I ask the developers there what they’re working on and what type of architectures they are implementing. The vast majority are (still) working with client/server architectures.

Performance is another binding thread in this and all of my books. But to me, “performance” means more than writing code that runs fast. It means writing code that helps developers perform more efficiently—write more solutions in less time and at a lower cost, not necessarily more code. This is especially important for developers who work in teams. All too often nowadays the programs we write were originally designed and implemented by others—or at least parts of them. This
means you have to deal with other developer’s designs, techniques, and documentation—assuming the latter exists. And others have to deal with what you write. We are constantly challenged to create code that can be passed on to others to integrate, support, or debug. There are a lot of techniques you can use to make this process easy—easy for you and for the people who have to deal with what you write.

This book, like my others, also discusses team development philosophy and how to best manage these teams. For example, in later chapters I mention that the development process requires what many programming shops, even the ones at Microsoft, sometimes don't have: discipline. No, I'm not talking about having a person walking the halls in leather and studs, cracking a whip (or at least I hope not)! I'm talking about establishing standards within the organization and sticking to them. It means writing specs and coding to spec. It means curbing your desire to code outside the spec because you know how to do something better than the toads that created the spec. That might be the case, but unless you get your co-workers and managers to help change the spec, what you’re creating is counterproductive. If you don’t code to spec, your innovative code is not likely to work with other developers’ code now or in the future. Discipline also means curbing your arrogance. Deciding that the consumer of your code can take it or lump it is hardly a productive attitude—but it is not an uncommon mindset in some shops. Of course, if you are paid by the line, you are free to ignore these suggestions. Just don’t look for repeat business.

I also look at ways to reduce COM overhead, reduce load time in your applications, and ways to better utilize the LAN, WAN, and Web, as well as RAM, disk space, and other system resources. I examine the newest .NET technology and show ways to make it work for you, instead of making more work for your server and staff. Most importantly, I look at how to write programs that other developers can understand and support—without you having to interrupt your Saturday afternoon barbecue.

The next few years are going to be very challenging for all of us. We’re all going to have to learn a new language and new design paradigms to stay competitive. If we don’t, we had better start looking for another way to earn a living. We’re going to discover that the new .NET Framework and the languages and tools that depend on it are full of new issues and challenges. Frankly, I’m excited about these challenges. I really like working with the new Visual Studio .NET and Visual Basic .NET. That’s the good news. The bad news is that once you’ve gotten over the initial shock and get used to the new features you, like I, will have trouble going back to Visual Basic 6.0, Visual InterDev, or Visual Basic Script. It’s like owning a Porsche and having to take a Ford Escort as your loaner car when your car is in the shop.