

Access 2002 Developer's Handbook

Danny Lesandrini

The *Access Developer's Handbook*, in all of its versions, is legendary. Danny Lesandrini takes a look at the latest version and let's you know why you need it.

Access 2002 Developer's Handbook

By Paul Litwin, Ken Getz, and Mike Gunderloy

Publisher: Sybex

ISBN: 0782140114

It seems that not a week goes by at the comp.databases.ms-access newsgroup without someone asking a question and being directed to the *Access Developer's Handbook (ADH)*. Invariably, the handbook thoroughly and completely answers the question, usually with a comprehensive explanation and the required code. If you've never seen a copy of the book, you owe it to yourself to give it a look. If, like me, you didn't read the Access 2000 version (I practically skipped Access 2000 entirely), you'll find that the sections on SQL Server, Data Access Pages, XML, and Web-related topics make the *Access 2002 Developer's Handbook* well worth the read (see [Figure 1](#)).

What is the ADH?

I'll be clear about what the ADH isn't: It's *not* an Access tutorial for beginners. The authors assume that the reader is familiar with the basics of creating tables,

queries, forms, and reports. Though macros are mentioned from time to time, you'll need at least a rudimentary understanding of how to write and run VBA code in standard modules to take advantage of the book. The handbook will teach you how to sharpen your existing VBA coding skills, and you'll learn some great tricks for increasing productivity while creating forms and reports. The insightful commentaries that accompany the practical, real-life examples include warnings to help you avoid common mistakes

As the title suggests, this is a handbook. Though you can read a handbook from cover to cover, as I did with the *ADH* in preparing this article, you probably won't. You're more likely just to familiarize yourself with where stuff is in the handbook so you can flip to the chapter you need when you need it. The introduction to the *ADH* says that the material was prepared in such a way that each chapter stands on its own, with cross-references to assist the reader in finding related information elsewhere in the two volumes. For me, a good programmer's handbook is a book that gets a place reserved for itself within arm's reach of my keyboard. The *ADH* meets all of these criteria.

What would a handbook be without examples and code snippets? The *ADH* includes examples that are concise and easy to follow. Each volume comes with its own companion CD, which includes every significant example in the text, as well as associated tables with sample data. Starting with the 2000 edition, the authors converted many of the "plug-and-play" generic procedures into Class modules for easy integration into your projects. If you're not accustomed to working with classes in Microsoft Access, then you'll really enjoy reading chapter three of the desktop handbook, "Using VBA Class Modules."

What's new: Desktop

Not having read the *ADH* for Access 2000, the line between "old" and "new" sometimes gets blurred for me, so what follows may more accurately titled, "What's new in 200x." Beginning with the Access 2000 *ADH*, the work spans two volumes. One volume



Figure 1. The two volumes that make up the *Access 2002 Developer's Handbook*.

focuses on desktop issues (more than 1,300 pages), while the other volume is about Access in the enterprise (nearly another 1,000 pages). Though I was most interested in enterprise issues, I read the desktop volume first, thinking I could breeze through it quickly. After all, I'd already read the Access 95 and Access 97 *ADHs*. How much new stuff could there be?

The answer: Plenty! Consider these new topics:

- Printer object and Printer collection
- PivotTable and PivotChart
- New report properties
- MDB conversion between version enhancements
- Office XP Web Components
- Form Dirty and Undo events

As I read these books, I kept two highlighters at my side: a yellow one for marking important points to remember and a red marker to flag warnings never to forget. An example of a point that merited yellow highlight is the discussion about when and why you'd use `DBEngine(0)(0)` instead of `CurrentDb()` to instantiate a DAO Database object. Though there's no simple answer, the *ADH* gives you the understanding you need to choose what's best for your needs. `CurrentDb()` is simpler and more intuitive but slower. `DBEngine(0)(0)` is faster, but it may not always be as up-to-date with changes to the user interface, such as the addition of new queries and tables to their respective collections.

I also tagged pages containing simple, but useful code (for example, how to retrieve values from controls on subforms and how to initialize list and combo boxes to display their first list item as selected). One point I marked with red highlighter was a detail about ADO recordsets that occasionally catches me unawares. When I open an ADO recordset using the `Execute` method of the `Command` object, I get an error when I try to add or edit records. This is because the `Execute` method always returns a read-only recordset. To create an editable ADO recordset, I must remember to use the `Open` method of the `Recordset` object. Someday, I'm going to remember this!

Oldie but goodie: Desktop

Covering new technology is, of course, interesting and sexy, but some of the most useful topics in the *ADH* have been there for years. Wherever possible, code previously written for the DAO library has been upgraded to work with ADO, the default data access method for Access 2000 and 2002. Some of the discussions in the book that I found most helpful included:

- The 20-step approach to database design
- Making form navigation behave like an Excel

spreadsheet navigation

- Printing mailing labels starting from a specific position
- Automatically resizing forms and their controls
- Tips for speeding up queries
- Tips for optimizing VBA code
- Tips for avoiding logic errors in your applications

I hazard to guess that regardless how long you've been working with Access, you'll find something new and useful in the chapters that discuss application optimization. The authors give great suggestions for making your applications run faster—or in some cases, just appear to run faster. Another well-represented topic is error handling and debugging. The 52 pages on error handling compose one of the best discussions I've read on the subject. Though more experienced developers won't likely find any new information here, the advice given serves as a good reminder of practices we should already be implementing.

What's new: Enterprise

In my opinion, programming with Microsoft Access really starts to get interesting as your applications move to the enterprise and to the Web. Accordingly, I was excited to finally begin reading the *Access 2002 Enterprise Developer's Handbook*. Although the distinction between desktop and enterprise topics is somewhat arbitrary, for the sake of this article (and the *ADH*), you can think of enterprise issues as those relating to handling many users using a client/server application, possibly using SQL Server and/or XML within a network, intranet, or over the Web.

Sure, it's fun and easy to scratch out a desktop application with Microsoft Access, but the real power in Access for the enterprise lies in its ability to compete with Visual Basic and ASP when creating business solutions. Access 2002 now includes even more tools to help you do this. The *ADH* covers these tools under the following topics:

- ImportXML, ExportXML and XSL generation
- SQL Server 2000 Desktop Engine
- Graphical designers for SQL Server stored procs and functions
- Data Access Page enhancements

The enterprise volume begins with a discussion on splitting your Access database into two: an application database and a data database. Both file server and true client/server scenarios are explained and compared. The chapter on Access security is the most comprehensive I've seen. In addition to describing how to implement security through the Access user interface, code is provided to manage

users and permissions using DAO, ADOX, and DDL SQL scripts.

Several chapters are devoted to working with SQL Server. The information included is sufficient to help an Access developer get started with SQL Server. Enough basic training is provided to show you how to create database objects such as tables, views, stored procs, and the like. Though a brief T-SQL tutorial is provided, if you plan to do anything fancy, you'll want to invest in a good book on SQL Server.

The last set of chapters deals with creating a Web interface for your Microsoft Access applications. Data Access Pages (DAPs) were first introduced in Access 2000 to help developers create Web pages for displaying and editing your Access data. Deploying DAPs was problematic in Access 2000, but, according to the *ADH*, the Access 2002 version is much more mature. In addition to explaining how to use the Access user interface to create deployable DAPs, the book includes a tutorial on creating basic ASP pages, even touching

on the new ASP.NET technology. One complaint: The chapter on integrating XML with Access 2002 covers all the technical aspects of importing and exporting data but lacks real-life scenarios where you might want to use this technology.

I usually like to end book reviews with a "What's the verdict?" section, but that question just doesn't apply to the *Access Developer's Handbook*. It's not a question of whether or not the book is a great value. There really is something for everyone in the *Access 2002 Developer's Handbook*. How much you personally can expect to leverage from these volumes depends on how much you may already know. For the serious Access developer, these books are a must-have. ▲

Danny J. Lesandrini, a Microsoft Certified Professional in Access, Visual Basic, and SQL Server, has been programming with Microsoft Access since 1995. He maintains a Web site containing Access-related code solutions at <http://datafast.cjb.net> and replies to all questions and comments sent to him via e-mail. datafast@attbi.com.

Access Efficiency...

Continued from page 9

down to the client. However, it's probably just as easy to reset the Recordsource as to reset the Server filter property.

So what should you do? First, if a form's response time is good enough, leave it alone! You have enough to do. Where there's a problem, stay away from the Filter property unless you absolutely need it. If you have a poorly performing form, make these changes:

1. Delete the entry in the form's Recordsource property.
2. Add a constant to your form that holds the SQL equivalent to the old Recordsource property.

```
Const strOldRecordSource As String = _  
    "Select * From Customers"
```

3. Wherever your code uses the Filter property, replace that code with code that updates the Recordsource property with your constant and the string that you used to put in the Filter:

```
Me.Filter = strOldRecordSource & " Where " & _  
    "...old text for the Filter property..."
```

If the original Recordsource already contained a Where clause, then you'll need to use this code:

```
Me.Filter = strOldRecordSource & " And " & _
```

"...old text for the Filter property..."

4. If you're using the Where parameter in the DoCmd.OpenForm method, move that parameter's entry to the OpenForm's last parameter (the OpenArgs parameter). In the Form_Load event update the Recordsource:

```
Me.Filter = strOldRecordSource & " And " & _  
    Me.OpenArgs
```

Alternatively, you could call me in to consult. My rates are very reasonable. In fact, call me in anyway—that way, you'll have someone to blame if this doesn't work out. ▲



[ACEFFF.ZIP at www.smartaccessnewsletter.com](http://www.smartaccessnewsletter.com)

Peter Vogel (MBA, MCSD) is the editor of *Smart Access* and a principal in PH&V Information Services. PH&V specializes in system design and development for systems that use Microsoft technologies. Peter has designed, built, and installed intranet and component-based systems for Bayer AG, Exxon, Christie Digital, and the Canadian Imperial Bank of Commerce. He's also the editor of Pinnacle's *XML Developer* newsletter and wrote *The Visual Basic Object and Component Handbook* (Prentice Hall, currently being revised for .NET). In addition to teaching for Learning Tree International, Peter wrote its Web application development, ASP.NET, and technical writing courses, along with being technical editor of its COM+ course. His articles have appeared in every major magazine devoted to VB-based development, can be found in the Microsoft Developer Network libraries, and will be included in Visual Studio .NET. Peter also presents at conferences around the world. peter.vogel@phvis.com.

Access 200 Developer's Handbook (Paul Litwin etc) Volume2: Enterprise Edition Chapter 2: Developing MultiUser Jet applications Page 32. Highlighted with 'WARNING'. John Howley. RE: In Access 2000 Developer's handbook, Ed2020 (Programmer) 11 Feb 02 06:54. Thanks John. I'd imagine the bug fix is called Access XP! ;o). Ed Metcalfe. RE: In Access 2000 Developer's handbook, Ed2020 (Programmer) 11 Feb 02 07:15. I tell a lie. The Nios® II Software Developer's Handbook assumes you have a basic familiarity with embedded processor concepts. You do not need to be familiar with any specific Intel FPGA technology or with Intel FPGA development tools. Familiarity with Intel FPGA hardware development tools can give you a deeper understanding of the reasoning behind the Nios® II software development environment. Accessing the processor is the same, regardless of whether a device implements only a Nios® II processor system, or whether the Nios® II processor is embedded deeply in a complex multiprocessor system. Therefore, you do not need to spend time manually creating interface mechanisms for the embedded processor. From Access 2 onward the Developer's Handbook series has proven indispensable. With thorough, clear presentations of essential details, excellent comparisons of previous versions to the newest version, genuinely useful examples (I've imported dozens into my projects), and plenty of advanced developer-oriented material, ADH is simply a must-have set. As a professional developer, you should have at least three such guides in your library. Access Developer's Handbook should absolutely be one of them. Most of them have been very helpful. The Access 2000 Developer's Handbook 2 Volume Set surpasses all others in terms of width, breadth, and depth of information (as well as in sheer weight). Many development reference books contain CDs with freebies, code examples, etc.