Tame the database monster!

Covers Access 2003, 2002, and 2000

FIXING

Access Annoyances™

"[O'Reilly books] are smart, concise, and on the bleeding edge."
—*Forbes*

Praise for Fixing Access Annoyances:

"A great reference for dealing with Access questions you always wanted to know the answers to but were too afraid to ask."
—*Kara Soos, Editor, Microsoft Access Fundamentals*

"If you're serious about Access, you must purchase this book."
—*Stephen Lebans, Access Developer and Microsoft Access MVP*

Phil Mitchell
& Evan Callahan

O'REILLY®

Access/Office Applications

FIXING Access Annoyances

Tame the database monster!

It's big, it's bad, and it runs your business. Access is a goulash of pitfalls and idiosyncrasies. Fear not—in your hands is the secret map to a treasure trove of fixes. Whether you're a database newbie or an advanced Access developer who's building a complex application, *Fixing Access Annoyances* is your one-stop shop for fixing what ails you.

About the Authors

Phil Mitchell has been developing database applications for over a decade, and is the lead analyst/programmer at the Harvard University and Harvard Medical School libraries, helping to invent the future of information and knowledge management. The most important thing he's doing these days is sustainability activism, as founder of the 2People project.

Evan Callahan actually worked for Microsoft, helping shape Access's manuals and Northwind sample database. Evan currently runs a consulting firm specializing in Microsoft Access solutions for small businesses, and is the author of several books on Access and Office.

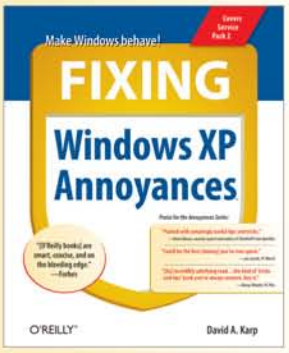
"There are few more venerable names in computer publishing than O'Reilly."—*Client Server News*

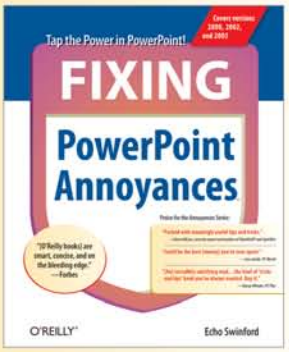
O'REILLY®
www.oreilly.com

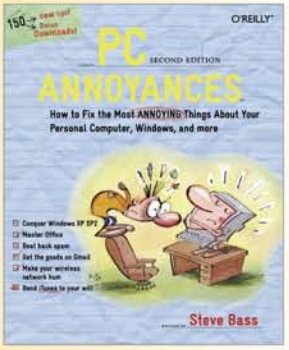
US $19.99 CAN $27.99
ISBN: 978-0-596-00852-9

Safari BOOKS ONLINE ENABLED Includes FREE 45-Day Online Edition

Recommended O'Reilly Expert Titles







FIXING ACCESS ANNOYANCES™

FIXING ACCESS ANNOYANCES™

How to Fix the Most Annoying Things About Your Favorite Database

Phil Mitchell and Evan Callahan

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

Fixing Access Annoyances™

How to Fix the Most Annoying Things About Your Favorite Database

by Phil Mitchell and Evan Callahan

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Print History:

February 2006: First Edition.

Editors: Michael Oliver-Goodwin and Robert Luhn

Production Editor: Genevieve d'Entremont

Copyeditor: Rachel Wheeler

Proofreader: Jeffrey Liggett

Indexer: Julie Hawks

Interior Designer: David Futato

Illustrators: Robert Romano, Jessamyn Read, and Lesley Borash

0-596-00852-X
[C]

♦ ♦ ♦

Dedicated to all those working toward an
ecologically sustainable and just future.

♦ ♦ ♦

Contents

INTRODUCTION **xv**

0. ACCESS BASICS **1**

The Database Window 1
Wizards 3
Design View and User Views 5
Setting Properties 6
Bound Versus Unbound Objects 7
Multi-User Databases and Split Design 8
How to Create an Event Procedure 9
How to Add Code to Your Application 10
DAO Versus ADO 11

1. GENERAL ANNOYANCES **13**

Customizing Access 14
Access's Bad Defaults 14
Alt-Tab Clutter 17
A Better Alt-Tab 18
The Recently Used File List 18
Grayed-Out Menu Items 18
Define Your Own Defaults for Forms, Reports, and Controls 21
Defaults for Tables, Queries, and Datasheets 22
Hidden (but Indispensable) Keyboard Shortcuts 22
Create Keyboard Shortcuts 24
Toggle Between Design View and Object View 26
Type Looooooong Lines in Small Places 26

Access Changes My Data 27
Help That Isn't/Is Helpful 28
Give Me Full Menus! 31
Missing or Mangled Toolbars and Menus 31
Missing Objects (Forms, Reports, Queries...) 32
Scroll Wheel Confusion 33
Turn Off Dire Hyperlink Warnings 34
Find/Replace for Database Objects 34
Save Custom Form Placement 35
Data Entry Without Access 35
Getting Lost in Datasheet View 36

Access Runs Amok 37
Access Is Buggy and Unstable 37
Flaky or Corrupted Database 38
That Darn Corruption 42
Missing Wizards or Import/Export/Link Options 44
Agonies of a Sick Installation 46

Error Messages 47
#Name? and #Error? 47
#Num!, #Div/0!, #Deleted, and #Locked 49
Enter Parameter Value 51
No Access License 52
User-Defined Type Not Defined 53
Missing References 54
Disk or Network Errors 55
Database Is Read-Only or File Is Already in Use 55

2. PERFORMANCE, VERSIONS, SECURITY, AND DEPLOYMENT 57

Performance 58
Database Bloat from Images or OLE Objects 58
Why Is My Database So Slow? 59
Why Is My Split Database So Slow? 64
XP Home Is Way Slow 65
Access with SQL Server Is Still Slow 65

Versions, Upgrades, and Conversion 67
Access Works on One PC but Not Another 67
Access/Windows Compatibility Issues 67
Converting Old Databases 68
Running the Same Database in Multiple Versions of Access 69
Running Different Versions of Access on the Same PC 69

Running the Right Version of Access 70
This Upgrade Is Killing Me! 71
Many Offices, Many Library Incompatibilities 72

Security 73
Access Security Is Weak 73
Upgrading Secured Databases 73
Database Lockout 73
Administering User-Level Security Correctly 75
Undo Botched Security 79
Frontend Secure, Backend Unprotected 80
First Login Fails After Securing Database 81
Too Many Logins 81
Splitting a Secured Database 82
Security Fails on Network Deployment 83
"Unsafe Expressions" Warning 83
Creating Digital Signatures 84

Enterprise Deployment and Networked Databases 86
Keeping Access Running Smoothly 86
Automating Maintenance Chores 87
Best Practices for Access on Networks 88
Getting Around Jet's 2-GB Limit 90
Avoiding Manual DSN Setup for ODBC 91
VB Code Fails After Split 93
Merging Two Frontends 93
Can't Get Replication to Work 95
Replication Fails on Frontend Objects 95
Corrupted Replicas 96

3. DATA, TABLES, AND DATABASE DESIGN 97

Database Design and Data Integrity 98
Table Design 101 98
Relationship Angst 100
I Don't Want to Design a Database from Scratch 104
Bad Field Names 105
Flaws in the Decimal Data Type 107
AutoNumber Nightmares 108
Can't Create a Relationship 109
Can't Delete Records 109
Default Values Don't Apply to Existing Records 110
Simple Validation Rules 112
Complex Validation Rules 114
Subtypes and Supertypes 115

Duplicate Records 117
Relationships Window Ghosts 117
Relationships Window Layout Distress 118
Attack of the Nonexistent Tables 119

Importing, Linking, and Exporting Data 120
Errors with Imported Data 120
Linked Spreadsheet Woes 121
Data Incorrectly Imported as Dates 123
Obscure Excel Import Errors 124
Unhelpful Error Messages During Import 124
Can't Import Word Tables 125
Copy/Paste to Excel Is Broken 126
Exporting Reports Produces Weird Numbers 127
Hidden Apostrophes in Exported Data 128
Exported Numeric Data Is Truncated 129
Exported Text Data Is Truncated 129
Miscellaneous Export Annoyances 130
Putting Data on the Web 131
Automating Import/Export 134
Exporting Data on the Fly 136

4. QUERIES 137

Query Basics 138
Query a Single Table 138
Query Multiple Tables 140
Meaningless Column Names 142
Create a Calculated Field 143
Create a Totals Query 144
Find the Properties Sheet for a Query 145
Divide and Conquer 145
Spurious Joins 146
Limit the Number of Records Returned 147
Avoid Duplicates in a Query 148
Find Duplicate Records 148
Count Yes/No Answers 149
Jump to SQL View 150
Speed Up Slow Queries 150

Query Misfires 152
Data Is Missing from a Multi-Table Query 152
Query Has No Data or Has Wrong Data 153
"Aggregate Function" Error 155

Totals Query Gives Incorrect Result 156
Sort Order Is Out of Order 157
Left Join Doesn't Work 157
"Join Expression Not Supported" and "Ambiguous Outer Joins" Errors 158
Input Mask Nixes Queries 159

Parameter and Crosstab Queries 161
Parameter Queries with Wildcards 161
Parameter Queries and Blank Responses 162
User-Friendly Parameter Queries 163
Parameter Queries That Accept Lists 165
Parameters in Crosstab Queries 167
Sorting Crosstab Rows Based on Totals 168
Crosstab Queries with Multiple Values 169

Techniques for Difficult Queries 171
Comparing Different Rows 171
Working with Hierarchies 172
Working with Ranges 174
Full Outer Joins 176
Divide and Conquer with Subqueries 176
Finding Rows That Don't Exist 178
Working with Calendar Dates 180
Missing SQL 183

5. FORMS 185

Working in Design View 186
Activating the Wizards 186
Edit Many Controls at Once 186
Leaving the Properties Sheet Open 187
Where Are the Form's Properties? 188
Attached Labels 188
Fix Combo and List Box Names 189
Subform Is Blank in Design View 190
Form Opens Slowly in Design View 190

Form Design 191
Building the Right Form 191
Synchronizing Subforms 194
Synchronizing Two Forms 195
Create Dialog Box Input Forms 197
Display Subform Values on the Main Form 199

Hide an Empty Subform 200
Reusing Subforms 200

Create Forms Your Users Will Love. 201
Simplify Data Entry 201
Open a Form to a New Record 204
Refresh Data Automatically 204
Hide Foreign Keys 205
Find Records Faster 207
Handle Items Not in a Combo Box 209
One Form, Many Screen Resolutions 211
Option Groups Don't Allow Text Values 214
Can't Change Column Headings for Datasheet View 216
Set Focus Doesn't Work 216
Enable Null Values in a Combo Box 217
Fix Scrolling in Combo Boxes 218
Use Arrow Keys in Combo Boxes 219
Speed Up Slow Combo Boxes 220
The Limits of Conditional Formatting 222
Form Is Blank in Form View 223
Edit User-Entered Data 224
Can't Save Changes to Data 224
Some Forms Shouldn't Be Editable by Default 227
Too Many Controls on One Page 228
Carry Data Forward to the Next Record 229

6. REPORTS, MAILING LABELS, AND CHARTS 231

General Report Annoyances 232
Report Wizard Isn't Working Its Magic 232
Build a Report from Scratch 234
Report Preview Is Too Small 236
Unexpected Parameter Boxes 237
Calculated Field Shows #Error? 237
Truncated Memo Field 238
Reports with No Data 238
Error 2501: The OpenReport Action Was Canceled 240
Filter with Parameters 240
Number Records Consecutively 243
Export to Word/RTF 244
Give Up and Try Excel Instead 246

Page Layout and Printing 247
Too Much Blank Space 247
Every Other Page Is Blank 249

Blank Page at End of Report 250
Column Layout Issues 251
At Least One Printer Required 252
Using the Default Printer 252

Sorting, Grouping, and Subreports 254
Records Aren't Sorted Like the Underlying Query 254
Orphaned Group Header 255
Subreport Loses Its Headings 255
Total Not Allowed in Page Footer 256
Crosstab Headings Don't Match 257

Mailing Labels 261
Off to See the Wizard 261
Custom Label Templates 261
Too Much Label, Not Enough Margin 262
Eliminate Blank Lines, and Other Label Tweaks 263
Prompt for Trim Function 266
Zip+4 Codes Run Together 266
More Than One Label for Each Record 266
Printing Partial Sheets of Labels 268

Charts and Graphs 269
Where's Charting? 269
Chart Only Shows Sample Data 270
Chart Changes Are Gone 272
Incorrect Data in Chart 273
Chart Is Blank 274
Fixed-Scale Charts 275
Control Charts in Code 275
Link a Chart from Excel 276

7. EXPRESSIONS, MACROS, CODE MODULES, AND CUSTOM CONTROLS 277

Expressions 278
.Dot, !Bang, and [Bracket] 278
Expressions That Go Blank 279
Tangled Up in Null 280
Debugging Expressions 282
Mysterious Syntax Errors 284
"Quotes", #Quotes#, and More #%&@!! """"Quotes""""" 284
Like, In, and Between Operators 286
"Like" Operator Changes to "Alike" 286
Expression-Building Blues 287

Dollar$ Sign$ Functions 288
[Brackets] Versus "Quotes" 289
Last Name, First 290
Dates! Dates! Dates! 290
Compare Dates, Ignore Times 294
Working Days Not Working 294
Refer to Subform Properties 295

Macros and Code Modules 296
Find a Macro Action 296
Tame the Visual Basic Editor 297
Find a Visual Basic Function 300
Pick the Right Event 301
Apostrophe Errors 301
Disable Confirmation Dialogs 302
Pause Macro or Code 303
Better Error Handling 305
Save a Record 308
Lost Data 308
Mysterious Syntax Errors, Part Deux 309

Custom Controls and External Applications 310
Mysteries of ActiveX Controls 310
ActiveX Control Is Missing Events 311
Calendar Controls 312
File Choosers 312
Use Excel Functions 315
Create Email Links 317
Send Email from Access 319

APPENDIX 321

GLOSSARY 339

INDEX 345

Introduction

"I have been at this for two months already. So far nothing seems to work."

—Seen in an Access newsgroup

The first time I used Access, I tore my hair out. I'd volunteered to help one of my favorite nonprofits (*http://www.earthdance.net*) fix their database, figuring that it would be a piece of cake for a "seasoned" pro like me. Instead, it was a nightmare. Every time I tried something that I knew *should* work, it didn't *quite* work.

Since that first wrestling bout, I've used Access many times in my job as a software developer. Once you know its pathways and tricks, it's almost indispensable—but it's as frustrating as it is useful. In fact, at one point when I was talking to O'Reilly Executive Editor Robert Luhn about doing such a book, he asked if Access had enough "annoyances." I shot back, "How many volumes did you have in mind?"

Access is a goulash of pitfalls and idiosyncrasies. Sometimes its user interface seems well designed and easy to use, and sometimes it seems to be a maze of narrow, twisty paths that dead-end. Some things in Access "just work," while others require a secret handshake. (Don't get me started on the Help system.)

For this book, I have been really fortunate to collaborate with Microsoft veteran Evan Callahan, who's been working with Access since Version 1.0. We've drawn on our own experience as developers and teachers, and we've scoured newsgroups, user groups, and online communities to identify the Access features, practices, and glitches that cause the most confusion, exasperation, and desperate calls to that Access guy your cousin Sharlene knows.

This book covers a wide range of annoyances, some so basic that even new Microsoft Office users will appreciate them, and some so advanced that only hard-core Access programmers will be able to follow the code. So before we start, let's talk about how you can get the most out of this book.

For openers, check out the Glossary at the back of the book. As you work your way through the various chapters, if you encounter a term or a concept that confuses you, check the Glossary. Hopefully you'll find a definition that helps.

For an overview of fundamental Access operations, see our cleverly titled "Chapter 0," *Access Basics*. For the inside scoop from Microsoft, see the section entitled "MSKB 123456" on how to mine the company's Knowledge Base articles. Finally, if you need more technical reference material (such as a list of Visual Basic functions or macro actions), check the Appendix at the back of the book.

How to Use This Book

This book starts out with general annoyances, and then targets specific areas such as queries and reports. Each chapter is organized (more or less) with simpler annoyances toward the front and more advanced material toward the back. If you open to one of the chapter-closers and find yourself neck-deep in complicated code, don't panic; the easy stuff is just a few pages toward the front. Also, we've tried to include enough background so that newbies can fix things they didn't even know were broken. If you're still working your way up the Access learning curve, there's a lot you can learn here.

There are two different kinds of annoyances in Access. One kind, such as those listed in "Errors with Imported Data" in Chapter 3, pops up and hits you over the head. You'll come to this book looking for help with these issues, because you can't get the @#$@%# features to work at all. The other kind of annoyance has to do with making Access work better (rather than just work at all). For example, you can use Access for months and never think of changing the default option that leaves Name AutoCorrect turned on (see "Access's Bad Defaults" in Chapter 1). Maybe no one ever told you that this is a hidden bomb that can cause database corruption and poor performance. When you start browsing through this book, pay special attention to the topics that offer preventive medicine (for instance, "Keeping Access Running Smoothly" in Chapter 2).

NOTE

Don't worry about typing in the long Visual Basic Applications (VBA) code samples you'll find in this book. They're available for download at http://www.oreilly.com/catalog/accessannoy/index.html.

Access Versions

Most of the material in this book is valid for most versions of Access, but when we need to be specific about commands or options, we usually refer to Access 2002 and 2003. We'll let you know if something is substantially different in Access 2000. We don't cover versions prior to Access 2000, but here and there we may drop hints for dealing with Access 97 hassles.

MSKB 123456

Throughout this book you'll see references that look like this: MSKB 209132. This is our shorthand for referring to Microsoft Knowledge Base articles by ID number. The main Knowledge Base is found at *http://support.microsoft.com/default.aspx?pr=kbhowto*, but to look up an article by its number you'll want to browse to Microsoft's advanced search page (*http://support.microsoft.com/search/?adv=1*) and set the Search Type drop-down menu to Article ID. In the For box above the Search Type drop-down menu, type in the number and hit Enter. You can also just do a plain search on the ID number, but you may have to wade through multiple hits.

If you're using a browser that supports keyword substitution in URLs (such as Mozilla or Firefox), bookmark *http://support.microsoft.com/default.aspx?kbid=%s* and set the keyword to "mskb". Then you'll be able to jump directly to any Knowledge Base article by entering "mskb" followed by the article ID in the browser's address box. (The browser substitutes the ID for the *%s* in the bookmark.) If, for some incomprehensible reason, you're still using Internet Explorer, there's a software patch that lets it do the same thing. See "A Shorter Path to Microsoft's Knowledge" in Steve Bass's *PC Annoyances*, Second Edition (O'Reilly) for details, and download PowerToys for Windows XP from *http://www.oreilly.com/catalog/pcannoy2/index.html*.

There's a wealth of useful information in the Knowledge Base, but the search interface has some maddening blind spots. If you're getting no hits on a specific phrase (and you really, truly spelled it correctly), chances are you set your search to use "The exact phrase entered." That may seem like a reasonable option, but it doesn't work very well. Instead, select "All of the words entered," and you should get the hits you're looking for. (Incidentally, Google crawls the Knowledge Base, so you can bypass Microsoft's search interface entirely by doing an advanced Google search and limiting your results to the *support.microsoft.com* domain.)

Most of the Knowledge Base's search interface is self-explanatory, but there are a couple of places where you need to be careful. First, you'll probably want to click the "Specify a product or version" link to limit your search to materials related to Access. We recommend searching for information on *all* versions of Access—you'll miss too many useful articles if you limit

your search to a specific version. To specify all versions of Access, go to the advanced search page noted above, pop open the Search Product drop-down, and select "More Products." Scroll down and click the "Office Access" item that has no version number.

The "Show results for" drop-down on the advanced search page is just as confusing. By default it's set to "This product only," which means that it ignores articles that aren't associated with the specific product you've selected (i.e., Access). This is usually fine, but if you're not getting any hits you should try "This product first," which will scour the whole Knowledge Base and use your product preference to weight hits accordingly.

Conventions Used in This Book

The following typographic conventions are used in this book:

- *Italic* is used for filenames, pathnames, URLs, email addresses, new terms where they are defined, and emphasis.
- `Constant width` is used for code excerpts, commands, method names, and function names.
- **`Constant width bold`** is used for items that should be typed verbatim by the user.
- ***`Constant width italic`*** is used for text that should be replaced with user-supplied values.
- Menu sequences are separated by arrows, such as Data→List→Create List. Tabs, radio buttons, checkboxes, and the like are identified by name, such as "click the Options tab and check the 'Always show full menus' box."

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Fixing Access Annoyances* by Phil Mitchell and Evan Callahan. Copyright 2006 O'Reilly Media, Inc., 0-596-00852-X."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at *permissions@oreilly.com*.

I Did Exactly What You Said, and It Still Doesn't Work!

This book is full of advice, and all of it has been tested. So how come some fix doesn't work for you? Well...there *could* be something about your setup or Access version that's causing problems. It's even possible that one of our pinpoint instructions somehow eluded your unflinching eye. Or maybe, just maybe, we screwed up. In any case, let us know! We want to hear from you. Send email to *annoyances@oreilly.com*.

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means it's available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books: it's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *http://safari.oreilly.com*.

Acknowledgments

First, we'd like to acknowledge the Access MVPs, and everyone who contributes to Access newsgroups; this is where so many of the "gotchas" in Access are brought to light, and these discussions were a great source of ideas for this book. Access MVP Lynn Trapp made an exceptional contribution to the early chapters of this book, and MVP Joan Wild kindly commented on an early outline. The Microsoft Support professionals who write the articles in the Knowledge Base also deserve our thanks for the expertise they share with the Access community.

Thanks to "Uncle Bob" (a.k.a. Robert Luhn, Executive Editor at O'Reilly), specifically for his prank phone calls pretending to be a foreign speaker with little English seeking Access help, and, in general, for shepherding this project through its highs and lows with unflappability.

Thanks also to Michael Oliver-Goodwin, editor extraordinaire, for his steamrolling patience and meticulous desire to make the book better. He didn't even lose his cool when we got cranky....

O'Reilly Would Like to Hear from You

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

There is a web page for this book, where you'll find links to the code and utilities mentioned in the book. You'll also find errata and additional information. You can access this page via:

http://annoyances.oreilly.com

To comment or ask technical questions about this book, send email to:

annoyances@oreilly.com

For more information about books, conferences, Resource Centers, and the O'Reilly Network, go to:

http://www.oreilly.com

Access Basics

0

If you're already an Access ace, you can skip this chapter and move right into the meat of the book. But if you're still clawing your way up the learning curve, here's a brief overview of some basic Access concepts and procedures.

The Database Window

When you start Access and create a new database (File→New), you'll find yourself in the *Database window*. This is home base when working in Access. A list of the object types in your database appears on the left side of the window. You can click an object name to show objects of that type (of course, there won't be any until you create some). The key types of objects are:

Tables

Tables are where data gets stored. A table in Access is roughly analogous to a table in Word or a worksheet in Excel—column headings (a.k.a. *fields*) and rows (a.k.a. *records*) are the main structural elements. The difference is that in a database, you must define these structures much more precisely. This is the foundation on which everything else rests. (See "Table Design 101" in Chapter 3 for more info.)

Queries

Queries which are based on Structured Query Language (SQL) typically retrieve selected data from tables. They can also be used to modify or delete existing data, add new data, and create new tables. Of course, you could view all of your data just by looking at the table, but the power of a database is that it lets you create alternate views of your data to answer specific questions. For example, your boss may say, "I need a list of all our products with sales over $10,000 per month that come from a single supplier." You'd go nuts trying to extract this information by looking at the raw data in a table, but you

can easily find it with a query. (See "Query Basics" in Chapter 4 for more information.)

Forms

As a general rule, you don't want users entering data directly into tables; there's too much potential for error and data damage. The safest and easiest way for users to enter data is via forms. You can use forms to provide clear, user-friendly data entry templates, and with a bit of Visual Basic (VB) code you can add an additional layer of error-checking to prevent erroneous or improperly formatted entries. Forms are also a useful way to browse a database and look up information. If you've ever filled out a web form, you're already familiar with the basic concept behind Access forms: text boxes, checkboxes, and other widgets provide ways for users to input data in a structured way. Access has a Form Wizard that makes it easy to create a basic form for any table. (For more on forms, see Chapter 5.)

Reports

Reports, which are actually based on queries, let you present, summarize, and print your data in various elegant ways. The key to a good report is basing it on a query that selects all the data the report needs from the underlying tables. Access's Report Wizard makes it easy to group, order, and summarize the data from the query. (We provide lots more information on reports in Chapter 6.)

Pages

Data Access Pages use Microsoft technology designed to let users view *and* edit Access data on the Web. Unfortunately, the technology doesn't work very well. At best, the feature is useful for intranets, but we recommend you look into other solutions first. (For more information, see "Putting Data on the Web" in Chapter 3.)

Macros

Much like macros in Word and Excel, Access macros can be used to automate certain tasks. For example, if you import the same worksheet data into a database every week, you can set up a macro that does the job with the click of a button. Unlike Word or Excel macros, however, you *cannot* create Access macros by recording a series of keystrokes and button clicks. Instead, you must construct a macro explicitly from a list of predefined actions, using the Macro Designer.

Modules

One of the most powerful aspects of Access is that you can write Visual Basic code to do everything you can do with macros, plus a whole lot more. Modules are where you store that Visual Basic code. If you're not a programmer, don't worry—you can accomplish a lot in Access without writing any code. But as you become more familiar with Access, you'll find that even a line or two of VB code can make life a lot easier.

We've provided helpful code samples in various chapters, along with an overview titled "How to Create an Event Procedure" that you'll find later in this chapter.

Wizards

If there's one secret to becoming instantly productive with Access, it's the wizards. These step-by-step dialog boxes walk you through many key tasks. Even advanced developers use them, simply because it saves time; the Form, Report, and Mailing Label Wizards are especially useful.

Here's a list of the main wizards. Unless otherwise noted, these wizards can be activated in the Database window by first selecting the specific type of object you want to create and then clicking the New button:

Form Wizard
: This wizard creates a basic, no-frills data entry form for any table. It's very useful. Power users will probably want to open the resulting form in Design View and tweak it (see Chapter 5), but the wizard gives you a good starting point.

Control wizards
: Forms, queries, and reports are based on *controls*—text boxes, labels, and so on. Access provides several wizards that make it easy to customize these controls and apply the kinds of tweaks that were once the preserve of power users. Here's where you can make a big improvement in the usability of your forms with surprisingly little effort (see "Activating the Wizards" in Chapter 5).

  For example, instead of using a text box in a form where users must enter country names, you could deploy the Combo Box Wizard to create a combo box control, providing a drop-down menu that lets users pick standardized country names from a list stored in a table. This prevents spelling errors and ensures that every user enters the same version of each name. Some other controls that have useful wizards include:

  Command buttons
  : The wizard makes it easy to put a button on your form that can trigger various actions, such as opening another form, running a macro, or saving the current record.

  List boxes
  : List boxes are similar to combo boxes, but instead of seeing a drop-down menu, users see a list of all the items at once.

  Option groups
  : Option groups present a fixed choice ("Yes," "No," "Maybe") as a set of mutually exclusive radio buttons.

Report Wizard

This wizard creates a basic, no-frills report based on a table or query. It's a good starting point, but you'll definitely need to go in afterwards and tweak the design. (For details, see Chapter 6.) Since reports are usually based on queries, designing a good report means building the right query first—many headaches in report design are due to so-so queries. Chapter 4 will put you on the path to understanding how to build the query you need.

Mailing Label Wizard

The Mailing Label Wizard has built-in knowledge of the sizes and specifications of many commercial labels and is an indispensable tool for creating mailing labels. Though there are some gotchas (see "Mailing Labels" in Chapter 6), it makes creating labels pretty easy. You can even create custom specifications for odd-sized labels. Since mailing labels are technically a kind of report, you invoke the Mailing Label Wizard from the New Report dialog (click Insert→Report from the Database window).

Chart Wizard

Building an Access chart from scratch can be painful, because there's no integrated graphical editor; every time you make a change, you must switch views to see the results. Though it's pretty basic, the Chart Wizard will at least get you started. For solutions to common problems, see "Charts and Graphs" in Chapter 6.

Table Wizard

This wizard gives you some sample prefab tables that you can customize. It's worth a look, but it's not especially useful. In most cases you'll want to build tables by hand. (See Chapter 3 for tips on designing your tables.)

Query wizards

The Simple Query Wizard enables you to build (what else?) a very simple query. This may be useful for absolute beginners, but it won't help much with most real-life queries. The other query wizards are more useful, but mainly for very specialized needs, such as crosstabs and duplicates queries. You'll need to design most queries yourself (see Chapter 4).

Design View and User Views

Wizards only take you so far. If you spend much time developing Access databases, most of the time you'll probably be working in *Design View*. Every type of object (table, query, report, and so on) has its own Design View. For example, in Table Design View you can define field names and determine what type of data is allowed to go into each field. In form and report Design Views, you'll tweak the behavior and layout of your controls. Queries have two different Design Views: the Query Builder provides a graphical user interface, while the SQL View lets you write queries in the SQL programming language. You'll find much more information about these different views in subsequent chapters.

Any view that *isn't* a Design View is a *user view*: user views show your objects as they will appear to users who are entering or viewing data. (Neither macros nor modules have user views—they're just code that works behind the scenes.) Table 0-1 lists the user views associated with some of the objects you'll be working with.

Table 0-1. Main user views

Object	Most common user views
Table	Datasheet
Query	Datasheet
Form	Single, Continuous, Datasheet
Report	Print Preview, Layout Preview

Here's an overview of the main user views:

Datasheet View
: When you look directly at the data in a table, Access presents it in Datasheet View, a column and row layout that looks a lot like an Excel worksheet. The columns are the fields, and each row is a record. By default, you get the same view when you look at the results of a query, since a query is basically a recipe for creating a "virtual" (temporary) table (see "Query Basics" in Chapter 4). A form can also be presented in Datasheet View—this is mostly useful for certain kinds of subforms (i.e., forms nested inside other forms). See Chapter 5 for more on subforms.

Single Form View
: When you want your form to present one record at a time, use Single Form View; it's the default.

Continuous Forms View
: When you want users to view a full page of records at the same time (this is sort of like looking at search engine results on the Web; you wouldn't want them presented one hit at a time), use either this view or Datasheet View. Continuous Forms View works best when each record is relatively small, so that many records can fit on one page. To turn it on in Design View, set Default View on the Format tab in the form's properties sheet to "Continuous Forms."

Print Preview and Layout Preview
: To see a report as it will appear in print, use Print Preview. For long reports, this may take some time to view. You can view the layout of a sample of records by using Layout Preview instead.

Setting Properties

In many places throughout this book, we'll tell you to look at, or set, this or that *property* of an object. Nearly all Access objects (tables, fields, queries, joins, forms, reports, sections, controls, and pages) have properties that define the way they look and work. For example, a form's Caption property determines the text that appears in the titlebar of that form's window.

In general, you will enter and change property settings in the object's properties sheet. However, the properties sheet isn't always found in the same way, and sometimes it's hard to find, period. Here are the steps for setting a property:

1. Open a table, query, form, or report in Design View.
2. Open the properties sheet for the object (for some of these objects, you can just press F4):

   Table
   : Click View→Properties.

   Query
   : Click the background of the upper pane of the query window, then click View→Properties.

   Table field
   : Click on the field; its properties sheet will appear in the bottom half of the window.

   Query field
   : Click the field in the query grid and click View→Properties.

   Query join
   : Double-click the join line between field lists.

Form or report
: Double-click the form or report selector box at the top-left intersection of the rulers.

Control or section
: Double-click the control or section bar.

Data Access Page
: Open the page in Design View, select View→Properties, and then select Edit→Select Page.

3. If the properties sheet organizes properties on tabs, click the tab that contains the property you want, or click the All tab to see all the properties in a single, scrollable list (see Figure 0-1).

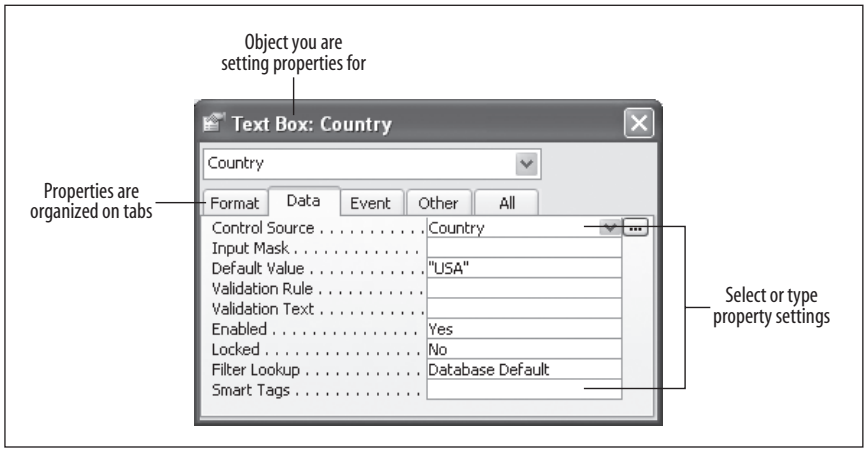


Figure 0-1. The properties sheet for a text box control.

4. Click the properties field you want to change and type in a setting. If there's a drop-down arrow, click it to select a property setting from the list. If a Build (...) button appears, click it to display an appropriate builder for the property.

Bound Versus Unbound Objects

You hear Access programmers make this distinction all the time, because it's fundamental to the way forms and controls work. An object is *bound* when it's linked directly to a data source (a table, say) and *unbound* when it isn't. The simplest example is a text box control on a form. Let's say you have a customers data entry form, with a text box where the user can type in the

company name. The text box's Control Source property (on the Data tab of its properties sheet) is set to the companyName field in the customers table. This ensures that when data is typed into the text box and saved, it goes directly into that field on that table; the text box is *bound* to that field.

On the other hand, you might have a data display form with two text boxes in which users type start and end dates, so the form will select and display a set of orders matching that date range. (Essentially, the text boxes allow the user to enter query parameters.) The Control Source properties of the two text boxes are blank, because the dates that get entered into them won't be stored permanently in the database; they are simply used to limit the range of the query. These text boxes are *unbound*.

There's actually a third case, too. Sometimes a text box has its Control Source property set to an expression—for instance, the total value of a customer's orders. This is called a *calculated* field, and it is neither bound nor unbound. Because the value of this field is calculated, you can't edit it or save it into the database. It's for display only.

In addition to controls, a form itself can be bound or unbound. A form is *bound* when its Record Source property (on the Data tab) is set to a table or query. If its Record Source is left blank, it is *unbound*. Generally, bound forms are used to display bound and calculated controls, and unbound forms are used to display unbound controls.

Multi-User Databases and Split Design

A *single-user database* lives on a single PC and is used by only one person at a time. This might be a personal contacts database, or an orders database that's used by several people who share the same computer. Obviously, sharing a single PC can be inconvenient, and that's why you can also use Access to design and manage multi-user databases.

A *multi-user database* can be accessed over a network by multiple users, from multiple PCs, at the same time. Access databases are often deployed this way, but it's important to do it correctly. Deploying Access across a network increases the risk of database corruption (see "That Darn Corruption" in Chapter 1), but if you do it right, the risk is manageable and the gain is substantial.

To create a multi-user database, it's standard to adopt a *split* design. *Splitting* involves separating your database into two different Access database (MDB) files: a backend that contains only your data (i.e., your tables) and a frontend that contains all of your user interface paraphernalia (que-

ries, forms, reports, and so on). This is easier to do than it sounds, because Access has a Database Splitter tool that makes it easy to split an MDB file into a backend and a frontend: it moves your tables into the backend and creates links to them from the frontend.

Once you've split your database, it's important to deploy it correctly. Because all users will share a single copy of the backend, it must be put in a network-accessible location—ideally, on a server. That way, all users can access the same data at all times. On the other hand, every user gets her own copy of the frontend, installed on her own PC.

It's a bad idea to attempt to use a split design over a wide area network (WAN) or the Internet; the reliability of the network connection is simply not good enough to prevent database corruption. One option is replication, which enables each user to have his own copy of the database, which is then synchronized periodically with the master copy. (See the sidebar "What Is Replication?" in Chapter 2 for more information.) However, we recommend against replication, because it's very difficult to implement correctly. An option we *do* recommend is to web-enable your database. (See "Putting Data on the Web" in Chapter 3 for a full discussion.)

How to Create an Event Procedure

We've already mentioned using the properties sheet to define the way controls look or work. That's just the beginning. Access has thousands of places where one or two lines of code added to an "event procedure" can fix a problem or dramatically improve an application. Throughout this book, we'll say things like: "Add this line of code to the Before Update event," or "Use the Click event of the command button to make this happen." Here's how it works.

At certain points in the program flow (when the user enters data, for instance, or runs a query), Access checks to see whether a developer has added any custom code. For example, every time a report is opened, Access sees if any code has been added to that report's Open event. If code *has* been added, Access runs it before it opens the report. This is powerful!

To cite a very simple example, you can automatically maximize a report window (see "Report Preview Is Too Small" in Chapter 6 for the full story) by dropping a single line of code in the report's Open event, such as:

```
DoCmd.Maximize
```

Here's how:

1. Open the report in Design View, and then open its properties sheet (View→Properties).

2. On the Event tab (see Figure 0-2), click in the On Open field, and then click the Build (...) button that appears. If you get a Choose Builder dialog, select "Code Builder" and then click OK. The Visual Basic Editor will open and create the appropriate empty event procedure for you. The empty procedure will look something like this:

```
Private Sub Report_Open(Cancel As Integer)

End Sub
```

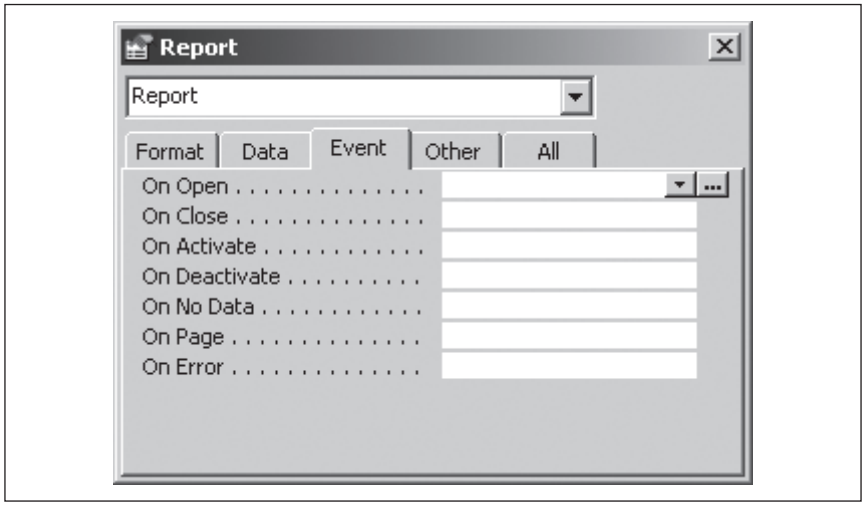


Figure 0-2. The Event tab shows the global report events—click in a particular field, then click the Build (. . .) button on the right to create an event procedure.

3. The procedure starts executing at the top and continues line-by-line until it gets to the bottom. Simply add your code on the blank line in between the top and bottom lines, so it looks like this:

```
Private Sub Report_Open(Cancel As Integer)
   DoCmd.Maximize
End Sub
```

   Indentation and blank lines have no effect on what your code does, but they do help to make code more readable.

That's all there is to it. Of course, the more you learn Visual Basic, the better you'll be able to take advantage of these event hooks. See the Appendix for a list of all Access events organized by category, and see "Tame the Visual Basic Editor" in Chapter 7 for help on using the VB Editor.

How to Add Code to Your Application

Adding a few lines of code to an event procedure on a properties sheet is easy. But suppose you want to add a *lot* of code?

There are numerous annoyances in this book where we say something like, "Fortunately you won't have to write this code yourself, because so-and-so has already written it. And here it is." But in order to take advantage of that free code, you must know how to add it to your application, and then how to run it.

The first part—adding the code—is easy, and that's what we'll cover here. The second part—running it—depends on the code. In some cases, you can run it immediately simply by typing the name of the new procedure in Visual Basic's Immediate window and pressing Enter; in others, you'll need to add at least a line or two of additional code yourself—often in an event procedure.

For now, let's talk about adding code to your database. For example, in Chapter 7 (see "Use Excel Functions") we show you how to call functions that are available only in Excel. Suppose you decide that the `xlCeiling` function is just what you need for your query. Here's what you'd do:

1. **Create a new model.** In the Database window, select Insert→Module. The Visual Basic Editor opens and puts the cursor into the new module.
2. **Add the code.** Copy the code in question (such as one of the samples we list on the *Fixing Access Annoyances* page at *http://www.oreilly.com/catalog/accessannoy/index.html*) and paste it into the new module. If you prefer, you can even type it in, but if you do this you are more likely to introduce typos.
3. **Save the module.** Ctrl-S saves the new module and lets you give it a name; something like **`basWhatever`** is good. ("bas" is the standard prefix for code modules—it stands for "basic," as in Visual Basic; **`Whatever`** is the memorable name you supply.)

That's all there is to it. Now you can use the `xlCeiling` function in expressions, just like any other Visual Basic function. For example, suppose your query showed grade averages, and you wanted to round them using the Excel `CEILING` function with a "significance" of 5. In the properties sheet of the Field line, instead of `avgGrade`, you could now write `xlCeiling([avgGrade], 5)`.

DAO Versus ADO

OK, you're ready to write some VB code. Prior to Access 2000, there was a single code library for working with data via recordsets, bookmarks, and so forth: the Data Access Objects (DAO) library. With Access 2000, however, Microsoft introduced a new code library: ActiveX Data Objects (ADO). Whereas DAO was designed specifically for Access's Jet database engine—and may actually offer some performance advantages with data in

MDB files—ADO is intended to work with a wider variety of data sources, including any Object Linking and Embedding (OLE) DB sources.

Does this mean that you can now forget about DAO and just work with ADO? Unfortunately, no. For example, the default recordsets that Access uses for bound forms are DAO objects. If you want to work with them, you must use DAO. In general, when you're working strictly inside MDB files (as opposed to using Access as a frontend to another Database Management System [DBMS], such as SQL Server), you may want to use DAO exclusively to avoid confusion. If you need to use a mix of DAO and ADO code, remember to be explicit about which library you are using. When you declare objects such as recordsets or connections, you must be sure to specify which type you mean, like this:

```
Dim rst1 As DAO.Recordset
Dim rst2 As ADODB.Recordset
```

Whichever library or libraries you use, make sure your database has references to them. In the VB Editor, click Tools→References and make sure Microsoft DAO Object Library and/or Microsoft ActiveX Data Objects Library are checked—use the latest versions. If you plan to use DAO or ADO only, uncheck the reference to the library you won't use to avoid confusion. Note that new databases created in Access 2003 include references to both libraries, while those created in Access 2002 and 2000 include only an ADO reference.

1 General Annoyances

It's fitting to start this book with a chapter on "General" Access annoyances, since no other military rank (Private? Major?) suggests how aggravating—or impressive—Access can be. Truly, Access is the supreme commander of all annoyances. And yet, when it works, it rules.

General annoyances aren't just annoyances we couldn't fit in anywhere else. They're global annoyances, such as the ridiculously bad default settings that affect everything you do in Access, mysterious `#Name?` errors that can crop up almost anywhere (in queries, forms, reports, and so on), and data corruption that can turn your database into digital mush. In this chapter we'll show you how to make the most of Access's user interface, demystify its obscure error messages, and take preventive measures against its worst booby traps.

CUSTOMIZING ACCESS

Access's Bad Defaults

THE ANNOYANCE: My database was running like a dog until I found out that Name AutoCorrect was turned on by default—and it's a known cause of corruption, too! Are there any other disasters hiding in Access's default settings?

THE FIX: You bet. Access has hundreds of default settings, and they range from quite sensible to pretty troublesome. Let's take a look at some of the worst offenders, along with some of our favorite tweaks. You can set the global defaults by clicking Tools→Options; the specific defaults that follow can be set inside individual forms or reports.

Global Defaults

Changing these global defaults will vastly improve Access's performance:

Turn off Name AutoCorrect.

When you change the name of a table (and many other things), Access is set by default to look for all the references to that table name and update them automatically. Although this can be convenient, it also causes a real performance hit. Worse yet, the feature has been known to cause database corruption. Just say no: select Tools→Options, click the General tab, uncheck the "Track name AutoCorrect info" box, and you're good to go (see Figure 1-1).

NOTE

Don't check the "Compact on Close" box. In this case, the (unchecked) default is right. It may sound like a good idea to have Access automatically compact your database every time you close it, but there's just one little problem: this feature can cause database corruption. Leave it unchecked.

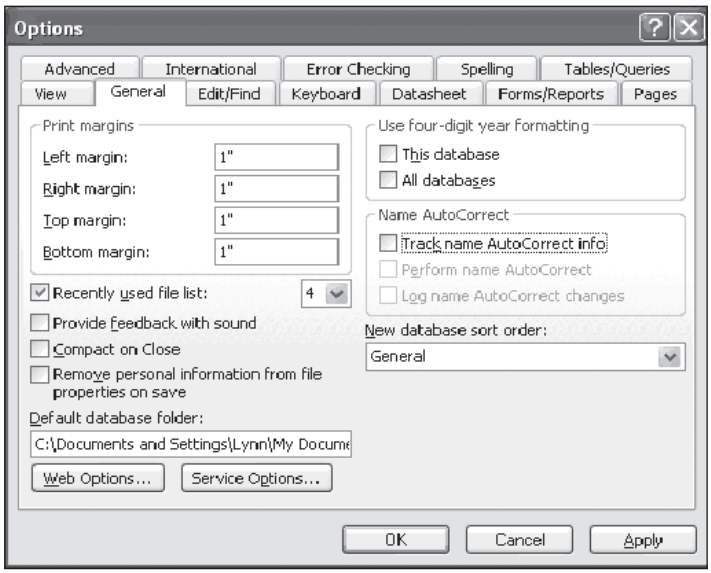


Figure 1-1. Turn off Name AutoCorrect by unchecking the "Track name AutoCorrect info" box.

Turn off spelling AutoCorrect.

You type in ACN (for the Association of Computing Nerds), and Access silently changes it to CAN, thinking it's correcting your typing. To avoid unexpected alterations to your data, turn off Office's spelling AutoCorrect. You can do this easily using Tools→AutoCorrect Options; see "Access Changes My Data," later in this chapter, for the details.

Reject AutoJoin.

A *join* is a temporary merging of two tables into a single virtual table. With AutoJoin on, Access looks for fields with the same name and automatically adds joins for those fields in the Query Designer. Depending on how you name your fields, this may be helpful, or it may produce lots of joins you don't want. In any case, it's easy enough to add them by hand. Also, if you create relationships between your tables, Access will add joins for them in the Query Designer automatically. So...turn off this automatic feature. Choose Tools→Options, click the Tables/Queries tab, and uncheck the "Enable AutoJoin" box.

Refine record locking.

If you're the only person using your database, you can skip this tip. But if multiple users are accessing the same MDB file at the same time, some form of record locking is essential to keep users from overwriting each other's data. The choices are found by selecting Tools→Options, clicking the Advanced tab, and looking in the "Default record locking" section. Access defaults to no locks, which means that records don't get locked until the moment they're saved. This can leave one user unable to save her edits if another user saves his changes first. A better choice is "Edited record," which locks the record at the moment it's opened for editing; multiple users are unable to edit the same record at the same time. Make sure you leave the "Open databases using record-level locking" box checked. Otherwise, page-level locking will be used, and users may be unnecessarily locked out of records that are *not* being edited by anyone. (In versions prior to Access 2000, you couldn't lock an individual record—you had to lock a 2-KB "page" containing the record in question, which meant locking neighboring records, too. Now that record-level locking is available, use it. It's a better choice.)

Stop unnecessary indexing.

Indexing is a technique Access uses to make search (and sort) operations faster; it relies on maintaining a pre-sorted copy of your data. (For more information, see "Speed Up Slow Queries" in Chapter 4.) With AutoIndex on, Access automatically creates indexes for fields whose names match a stored list ("id," "code," and so on). However, most of these indexes aren't really necessary; they slow Access's update performance, and they may cause you to exceed Access's 32-indexes-per-table limit. Access automatically indexes the primary key and unique fields anyway, and any other indexes you need can be added by hand. Click

Tools→Options, click the Tables/Queries tab, and clear the list of field names in the "AutoIndex on Import/Create" box.

Clean up your taskbar.

Tired of having every single database object you open show up in your Windows Taskbar? Turn off this annoying option by choosing Tools→Options, clicking the View tab, and unchecking the "Windows in Taskbar" box. Then multiple windows will be displayed as a drop-down list inside one single Taskbar item. (See "A Better Alt-Tab," later in this chapter, for alternatives.)

Specific Defaults

Here are some defaults you can set inside forms, reports, or tables to further improve performance:

Keep the Tab cycle inside one record.

By default, the Tab key moves the insertion point from field to field on a form; when it hits the end of the form, it moves on to the next record. This is great for expert data entry clerks, but confusing for inexperienced users. To change this behavior, open your form in Design View, open its properties sheet (View→Properties), select the Other tab, and set Cycle to "Current Record."

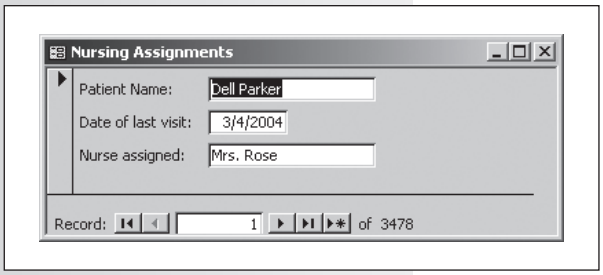


Figure 1-2. The record selector (bold black arrow) is not useful when your Form View shows only one record at a time.

Turn off record selectors.

Record selectors are the black arrows that Access puts to the left of a record, indicating that it's currently selected or being edited. They're useful in Continuous Forms or Datasheet View, where multiple records are displayed. However, if most of your forms display only a single record at a time (see Figure 1-2), you may want to turn these off. Open your form in Design View, open its properties sheet, go to the Format tab, and set Record Selectors to "No."

Disable design changes during data entry.

By default, design changes can be made to forms while they're open in Form View. This can be confusing to users, particularly if a properties sheet appears because it was left open in Design View. In most cases, you're better off restricting design changes to Design View. To do so, open your form in Design View, open its properties sheet (View→Properties), select the Other tab, and set Allow Design Changes to "Design View Only."

Allow users to save form layouts.

By default, no matter how you resize or lay out form windows, Access will not save your settings. This is more than a tad annoying, so open your form in Design View, open its properties sheet (View→Properties) and on the Format tab set Auto Resize and Auto Center to "No." See "Save Custom Form Placement," later in this chapter, for more details on saving form layouts.

Prevent shrunken reports.

By default, Access shrinks every report to fit the whole page on the screen, giving most users a single, unreadable window. To change this behavior, open the report in Design View, open its properties sheet (View→Properties), click the Format tab, and set Auto Resize to "No."

Disable the Subdatasheet Name property.

If you have any one-to-many relationships in your database (e.g., customers-to-orders, where there is exactly one customer for each order but there can be many orders for a given customer). Access considerately adds subdatasheets to the tables on the "one" side of the relationship. (If "one-to-many" is an unfamiliar concept, see "Relationship Angst" in Chapter 3 for an explanation.) When you open your customers table in Datasheet View, you'll see a little plus sign next to each record; click to open the related records from the "many" side of the relationship.

This may sound helpful, but to provide this service Access must constantly check to see if these relationships exist, which drags down overall performance. To prevent this from happening, open your table in Design View, open its properties sheet (View→Properties), and set Subdatasheet Name to "[None]." (If you don't want to do this for every table, see "Defaults for Tables, Queries, and Datasheets," later in this chapter.) For tables where you *do* want a subdatasheet, it's perfectly fine to set the Subdatasheet Name property to the specific name of a related table; just don't set it to "Auto."

Default Values for Numeric Fields

Unlike all the other data types, which by default have no values (i.e., are null), Access automatically sets the Number data type to default to zero. This is dumb. Although zero can be a useful default value, it shouldn't be assigned automatically—that's what null is for. Unfortunately, there's no way to clear this setting globally. Whenever you use the Number data type you'll have to clear its Default Value property manually.

Alt-Tab Clutter

THE ANNOYANCE: I like to use Alt-Tab to switch between open applications, but when I use Access, every single database object drops its own icon into the Windows Taskbar, so Alt-Tabbing means swimming through Access objects. Whose idea was this? Is there no end to this clutter?!

THE FIX: I guess it seemed like a good idea at the time. Fortunately, there's a simple fix: in Access, select Tools→Options, click the View tab, and uncheck the "Windows in Taskbar" box. This will give you a clutter-free Alt-Tab experience. To flip through Access objects and databases within Access, press Ctrl-F6.