Access™ 2007 VBA Programming FOR DUMMIES®

by Joseph C. Stockman and Alan Simpson

Wiley Publishing, Inc.

Access™ 2007 VBA Programming FOR DUMMIES®

Access™ 2007 VBA Programming FOR DUMMIES®

by Joseph C. Stockman and Alan Simpson

Wiley Publishing, Inc.

Access™ 2007 VBA Programming For Dummies®

Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada

Library of Congress Control Number: 2006939596

ISBN: 978-0-470-04653-1

Manufactured in the United States of America

Dedication

Joe Stockman: To my mom and all my friends and family who supported me — and left me alone — during this project.

Alan Simpson: To Susan, Ashley, and Alec, as always.

Authors' Acknowledgments

Even though only two authors' names appear on the cover, every book is a team project. These authors would like to thank the many people who contributed to this book. To Carole McClendon and everyone at Waterside Productions, thank you for finding this project and making it happen. Also, many thanks to Kyle Looper and Jean Rogers at Wiley for taking a chance on a new author to help with the rewrite. And also, thanks to Microsoft for making Access a wonderful development environment.

About the Author

Joe Stockman is an independent consultant, software designer, and author who has been using Microsoft Access since its initial release. He's also developed courseware and taught classes in Access and VBA. Joe developed his first application in Access, and then migrated into Visual Basic and VB.NET, where he specializes in creating applications for the Windows Mobile platform. He worked for several software companies before forming his consulting business in 2002, where he deals with all types of clients including healthcare, financial, government, manufacturing, and small business. His ability to turn his customers' wishes into working applications keeps them satisfied. Joe's also writing the fundamentals column for the *Advisor Guide to Microsoft Access* magazine.

Alan Simpson is the author of over 100 computer books on databases, Windows, Web site design and development, programming, and networking. His books are published throughout the world in over a dozen languages and have millions of copies. Alan has also taught introductory and advanced computer programming courses at San Diego State University and the UCSD Extension. He has served as a consultant on high-technology, education-oriented projects for the United States Navy and Air Force. Despite that, Alan has no fancy job title because he has never had a real job.

Publisher's Acknowledgments

We're proud of this book; please send us your comments through our online registration form located at www.dummies.com/register/.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Associate Project Editor: Jean Rogers

(Previous Edition: Christopher Morris)

Acquisitions Editor: Kyle Looper

Copy Editor: Becky Whitney

Technical Editor: Russ Mullen

Editorial Manager: Kevin Kirschner

Media Development Specialists: Angela Denny, Kate Jenkins, Steven Kudirka, Kit Malone

Media Development Coordinator: Laura Atkinson

Media Project Supervisor: Laura Moss

Media Development Manager: Laura VanWinkle

Media Development Associate Producer: Richard Graves

Editorial Assistant: Amanda Foxworth

Sr. Editorial Assistant: Cherie Case

Cartoons: Rich Tennant (www.the5thwave.com)

Composition Services

Project Coordinator: Jennifer Theriot

Layout and Graphics: Carl Byers, Stephanie D. Jumper, Barbara Moore, Barry Offringa, Alicia B. South

Proofreaders: Techbooks, Brian H. Walls

Indexer: Techbooks

Anniversary Logo Design: Richard Pacifico

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Joyce Pepple, Acquisitions Director

Composition Services

Gerry Fahey, Vice President of Production Services

Debbie Stailey, Director of Composition Services

Contents at a Glance

Table of Contents

Introduction

Welcome to *Access 2007 VBA Programming For Dummies.* As you already know (we hope), Microsoft Access is a huge database management program, offering lots of ways to manage data (information). Common uses of Access include managing mailing lists, memberships, scientific and statistical data, entire small businesses, and just about anything else that involves storing and managing large amounts of information.

As the title implies, this book is about using Visual Basic for Applications (VBA) to enhance the power of Access databases. If you want Access to print words on a check, skip mailing labels that you've already used, or manipulate data behind the scenes, you have to write VBA code.

By the time you finish this book, you should know exactly what VBA is all about and how it fits into Access. You'll discover the meanings of all those obscure terms that programmers throw around — code, variable, array, loop, object — as though they were common knowledge. You'll be able to write and use your own, custom code, just like advanced programmers do.

This book covers VBA in Access 2007. Although many changes and improvements to Access have occurred in all the versions that Microsoft has released, the VBA programming language has hardly changed a bit over the years. Although Access 2007 looks completely different from previous versions, the underlying objects are virtually unchanged. The code that you see in this book should also work in Access 2000, 2002, and 2003. The vast majority of the code in this book also works just fine even in last century's versions, such as Access 97.

About This Book

We wish we could say that this book is exactly like a coffee-table book, where you could just pick it up, flip to any page, and have everything make perfect sense to you. Well, we *could* say that, but we'd be lying if we did. It's not because we want to break from the coffee-table book idea. It's really more because some stuff in life doesn't make much sense until after you already know something else.

Here, it isn't really possible to make much sense of VBA code until you understand what VBA code is and why it exists. And, we are talking about Microsoft Access VBA here. To make sense of much of anything in this book, you have to already be familiar with Microsoft Access tables, queries, forms, and reports. We just don't have enough room in this book to explain all that stuff from scratch and still have enough pages left over to talk about VBA.

On the bright side, we did everything we could to make it easy to find what you need to know, when you need to know it. You certainly don't have to read this book from cover to cover to make sense of things. After you find the topic you're looking for, you should be able to read through the section and be done with it quickly. Often, you can skip reading sections altogether and get all you need to know from looking at the figures.

Conventions Used in This Book

While we're on the topic of using this book without boring yourself to death by attempting to *read* it, we also stuck with some conventions for displaying text in these pages. For example, any VBA programming code appears in a `monospace` font with a gray background, like this:

```
'VBA code to say Hello World on the screen.
Sub Hello()
   MsgBox "Hello World"
End Sub
```

When we have just a little chunk of code to show in text, like this — `Dim Wit As Date` — you can see what is and what isn't VBA code.

The ➪ symbol that you see in text separates individual menu options (commands) that you choose in sequence. For example, rather than say "Choose New from the File menu" or "Click File on the menu bar and then click New on the drop-down menu," we just say

Choose File➪New from the menu bar.

When you see something **in bold,** we want you to enter (type) it.

What You're Not to Read

Not many people in the world would put reading a computer book into the Read for Fun category. We think that reading a computer book is more likely to fall into the Read for Work or Don't Read category. To minimize the time

you have to spend away from the fun stuff, we put some information in sidebars and beside Technical Stuff icons. That information is definitely optional reading that you're welcome to ignore.

Foolish Assumptions

To stay focused on VBA in this book, we need to assume that you're already familiar with Access and that you're comfortable creating tables, forms, reports, and queries. However, we don't assume that you're a true Microsoft Access expert. Let's face it: Access isn't exactly an easy program for most people to tackle.

Another assumption we make is that you have already created an Access database with at least some tables and forms in it. In fact, writing VBA code is usually the last step in creating a custom Access database.

Finally, we don't assume that you're already an accomplished programmer who is just picking up a new programming language. Rather, we assume that you've never written any programming code in your life — and maybe you aren't even all that sure what programming code means or how it relates to Microsoft Access.

How This Book Is Organized

All books contain a lot of information. That's what makes them books. To break down topics into smaller, more manageable chunks, we split this book into six main parts.

Part I: Introducing VBA Programming

This part has all the information you need to get started. If you've already been using VBA for a few months or years, you can skim this part. If you don't know a VBA procedure from a PTO meeting, you might want to take a closer look at Part I before venturing forth to other parts.

Part II: VBA Tools and Techniques

Here you discover how to write VBA code to make Access do things for you. For example, you'll see how you can make Access open forms, respond to button clicks, change the appearance of objects, and more.

Part III: VBA, Recordsets, and SQL

Here you get friendly with tools and techniques for managing your Access tables by using VBA with SQL (Structured Query Language) and recordsets. All those buzzwords make this process sound more technical than it really is. But as you'll see, if you've done anything at all with queries, you've already been working with SQL recordsets. The idea is the same. We just use fancier terminology in the VBA world.

Part IV: Applying VBA in the Real World

In this part, you get into some more advanced programming tricks, mostly by using techniques presented in earlier parts in new and creative ways. You'll also see how to use the VBA debugging techniques, which can be real life-savers when things go wrong and you just can't figure out why the code you wrote isn't doing what you intended.

Part V: Reaching Out with VBA

VBA isn't a programming language solely for Microsoft Access. You can also use VBA to customize all the Microsoft Office application programs, including Word, Excel, and Outlook. Furthermore, VBA can import data from, and export data to, a variety of formats that extend its reach even beyond Microsoft Access. Part V shows you how that's all done.

Part VI: The Part of Tens

What *For Dummies* book would be complete without a Part of Tens? Ten is such a nice number to work with, given our ten fingers and all. Chapter 15 covers the main strategies that you can adopt to avoid going crazy trying to get VBA to do your bidding. Chapter 16 goes over the top ten nerdy programming tricks you're most likely to want to do almost from your first day of using VBA.

Icons Used in This Book

As you flip through this book, you'll notice little icons sprinkled throughout its pages. These icons, as described here, point out little chunks of text that deserve either a little extra attention or very little attention:

Tips point out handy tricks or techniques that can make things easier for you when you're working with VBA.

These icons point out techniques that, if you do things wrong, might create problems. If you pay attention to the Warnings we give, you can avoid making common blunders.

These icons point out tools and techniques that you'll use often as you work with VBA. Keep them in mind.

These icons point out text that describes how or why a thing works the way it does from a technical standpoint. If you just want to get a thing to work and don't care about how or why it works, you can always skip these.

Web Site for This Book

If you can find a way to copy and paste — rather than type — VBA code into your database, go for it. Much of the sample VBA code shown in this book is the kind of thing you can just drop into an Access database and start using. There's no need to retype the whole thing. Anyway, we post all the useful code at this Web site:

```
www.dummies.com/go/access2007vbaprog
```

When you get to the site, you'll see where to find the code and how to copy and paste it into your own database, and find a link where you can send us your questions.

Where to Go from Here

Now that you know what this book is about and how it's organized, the next question is "Where do I start?" Your best bet, if you're an absolute VBA beginner, is at Chapter 1. Try to slog through the first three (short) chapters to get your bearings.

Experienced VBA users can probably start anywhere that looks interesting. If you get in over your head at some point, watch for cross-references to earlier chapters where you can quickly fill in the knowledge gap that's causing the confusion.

Part I

Introducing VBA Programming

The 5th Wave By Rich Tennant

"Once I told Mona that Access was an 'argument' based program, she seemed to warm up to it."

In this part . . .

VBA lets you do some pretty amazing stuff in an Access database. With VBA, you can make Access do boring, repetitive jobs that you might otherwise have to do on your own. You can even get Access to do things that it couldn't possibly do on its own. Before you dive right in and try to make such things happen, you need to step back a moment and get a feel for how VBA fits into the whole Microsoft Access scheme of things. Then you need to get friendly with the tools available to you for turning ideas into stuff that actually happens when you want it to happen. We lead you through all of that in Chapters 1 and 2.

With your road map and tool kit in hand, you'll be ready to get into what Access VBA is really all about — writing code (also known as programming) — to make Access do exactly what you want it to do. Yes, you write code by simply typing it, unless, of course, you can just copy and paste the code, as is often the case. Chapter 3 talks about both writing and swiping VBA code.

Chapter 1

Where VBA Fits In

In This Chapter

- Describing Access
- Discovering VBA
- Seeing where VBA lurks
- Understanding how VBA works

This book is about using *Visual Basic for Applications (VBA),* which is a programming language that helps you program, tweak, and squeeze productivity from Access. VBA, which is embedded in Access, is a sophisticated set of programming tools that you can use to harness the power of a packaged application like Access. Just like you need to know how to walk before you can run, you need to know how to use Access before you can start to use Access VBA.

Maybe you want to use Access to manage a large mailing list. Maybe you need Access to manage your whole business, including customers, products, and orders. Perhaps you need to manage enrollments in courses or events. Whatever your reason for using Access, your first step is always to create the tables for storing your data. From there, you can then create queries, forms, reports, and macros to help manage that data. All these steps take place before you even get into VBA, so in this book we have to assume that you're already an experienced Access user who needs more than what queries, forms, reports, and macros can provide. If you're new to Access, this book isn't a good place to start. If you need to brush up on Access, *Access 2007 For Dummies* (by John Kaufeld, Laurie Ulrich Fuller, and Ken Cook; Wiley Publishing) or *Access 2007 All-in-One Desk Reference For Dummies* (Alan Simpson, Margaret Levine Young, and Alison Barrows; Wiley) is a good place to start.

Although Access has progressed through many versions over the years, VBA has remained relatively unchanged. We used Access 2007 to create this book, but the code examples we present should work fine in just about any version of Access. So now, before launching into VBA, take a moment to delve into what tables, queries, forms, and reports are all about, and how VBA fits into the overall scheme of things.

Taking a Look at Access

Access, part of the Microsoft Office suite, is a huge database management system that you work with by using modern object-oriented methods. (The term *object-oriented* stems from the fact that everything you create in Access — a table, form, report, or whatever — is considered an object.

The Access Navigation pane, as shown in Figure 1-1, is the main container in which you store all the main objects that make up a single database. The Navigation pane breaks down the objects into groups — tables, queries, forms, and so on — and each group contains the objects within that group. The following list summarizes the types of objects.

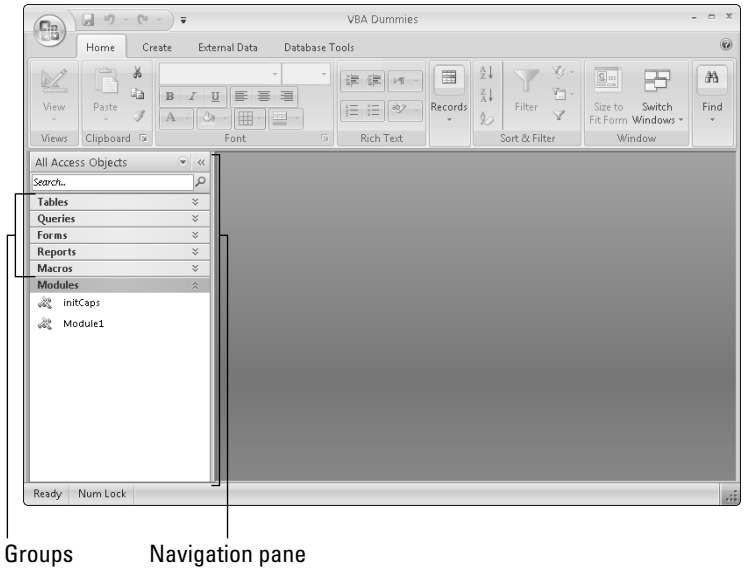

Figure 1-1: The Access Navigation pane.

- **Tables:** *Tables* contain the raw data that all other object types display and manage. Data in tables is stored in *records* (rows) and *fields* (columns).
- **Queries:** Use *queries* to sort and filter data from one or more tables.
- **Forms:** Access *forms* are similar to printed fill-in-the-blank forms, but they allow you to view and change data stored in Access tables.
- **Reports:** *Reports* define how data should be presented on printed pages.
- **Macros:** *Macros* provide a means of automating certain aspects of Access without programming in VBA.
- **Modules:** The *Modules* group, as you soon discover, is one of the places where you store VBA code. If you're not already familiar with modules, that's fine. Modules are what this book is really all about.

One of the most important things to understand is that you don't use VBA "instead of" other objects, like tables and forms. You use VBA to *enhance* the capabilities of other object types. Therefore, it makes no sense to even try VBA until you have a firm grasp of the purpose and capabilities of those other object types in Access.

Understanding VBA

Visual Basic is a programming language — a language for writing instructions that a computer can read and process. VBA is a programming language that's specifically designed to work with the application programs in Microsoft Office including Word, Excel, Outlook, and, of course, Access.

When you write text in a programming language (as opposed to writing in plain English), you're writing *code.* Programmers use the term *code* to refer to anything that's written in a computer programming language. For example, Figure 1-2 shows some sample VBA code. The whole trick to mastering VBA is finding out what all the various words in the language mean so that you can write code that tells Access exactly how to perform a task.

```
Public Function PCase(anyText)
    'Custom Access VBA function to fix all uppercase letters.

    PCase = StrConv(anyText, vbProperCase)

    If Left(PCase, 4) = "P.o." Then
        PCase = "P.O." & Mid(PCase, 5)
    End If

    If Left(PCase, 2) = "Mc" Then
        PCase = "Mc" & UCase(Mid(PCase, 3, 1)) & Mid(PCase, 4)
    End If

    If Left(PCase, 3) = "Mac" Then
        PCase = "Mac" & UCase(Mid(PCase, 4, 1)) & Mid(PCase, 5)
    End If

End Function
```

Figure 1-2: Some sample VBA code.

If the sample code shown in Figure 1-2 looks like meaningless gibberish to you, don't worry about it. People aren't born knowing how to read and write VBA code. Programming (writing code) is a skill you have to learn. For now, it's sufficient just to know what code looks like. Knowing what the code means is one of the skills you master in this book.

Because VBA code looks like a bunch of meaningless gibberish typed on a sheet of paper, it begs the question of why anybody would want to figure out how to read and write a dreadful language like that one. The answer to that question lies in the role that VBA plays in an application like an Access database.

Do, not die

Think of the term *execute* in the sense of "to carry out," as when you execute a U-turn or execute a procedure. Don't think of *execute* in the sense of "terminate the life of."

Access does indeed have a ton of tools that let you create a database without any programming. You could easily spend months or years just finding all the things you can do in Access without writing any VBA code. Yet despite the huge number of things you can do without programming, sometimes you want your database to accomplish a task that's not built into Access. That's where VBA comes in. When you want Access to perform a task that it doesn't already know how to perform, you write the steps to be performed in the VBA programming language.

When you're writing VBA code or just looking at some VBA code written by someone else, Access doesn't do anything. Access doesn't start performing the steps described by that code until Access executes the code. When you write VBA code, you're writing a set of instructions that Access can perform at any time, over and over again.

The ability to use the same code over and over again is the key to automating mundane tasks in Access. For example, if you use Access to print checks, you might have to manually type the part of the check where you spell out the amount, like "Ninety-two and 99/100 Dollars" for $92.99 because Access can't make that translation on its own. But if you could write some code to translate a number like $92.99 into words, you wouldn't need to type all those dollar amounts. Access would just print the correct information as it prints each check.

Seeing Where VBA Lurks

In an Access database, VBA code is stored in modules. Despite its fancy name, a *module* is basically an electronic sheet of paper on which VBA code is typed. A module in Access is either of these two types:

- **Standard:** A page that contains VBA code that's accessible to all objects in the database. A standard module always exists in the Modules group in the Navigation pane.
- **Class:** A page of VBA code that's attached to every form and report you create. You can also create a class module that appears in the Navigation pane.