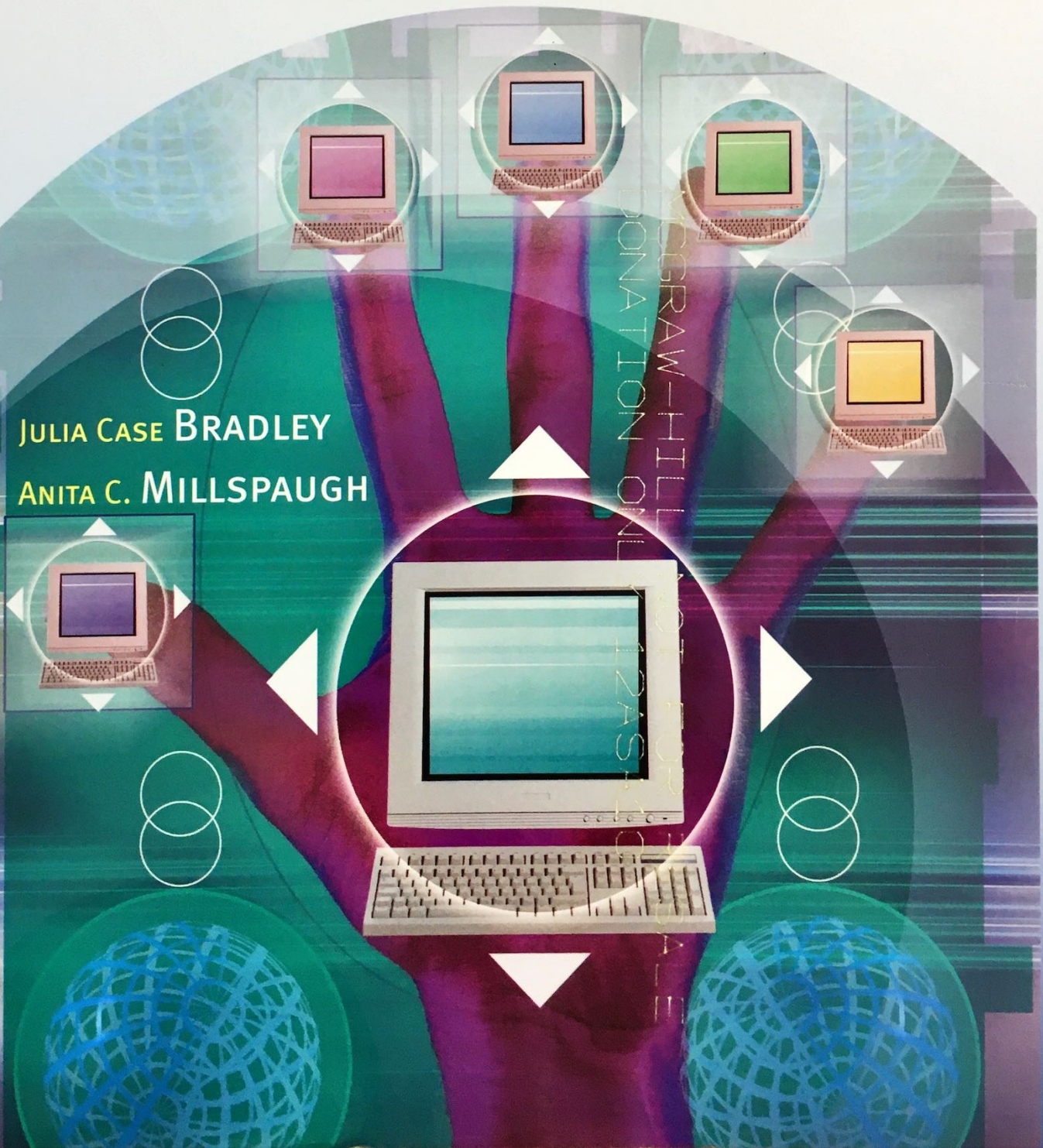


Advanced Programming Using Visual Basic 2005

JULIA CASE BRADLEY
ANITA C. MILLSPAUGH

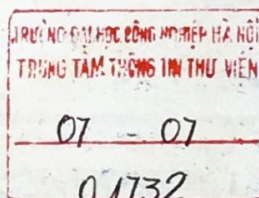
PROGRAM-HILL
DONATION ONLY
122AS-110



ADVANCED PROGRAMMING USING VISUAL BASIC 2005

Julia Case Bradley
Mt. San Antonio College

Anita C. Millspaugh
Mt. San Antonio College



McGraw-Hill
Irwin

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Kuala Lumpur Lisbon London Madrid Mexico City
Milan Montreal New Delhi Santiago Seoul Singapore Sydney Taipei Toronto



ADVANCED PROGRAMMING USING VISUAL BASIC 2005

Published by McGraw-Hill/Irwin, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY, 10020. Copyright © 2007 by The McGraw-Hill Companies, Inc. All rights reserved. Previous editions © 2001 and 2003. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

2 3 4 5 6 7 8 9 0 QPD/QPD 0 9 8 7

ISBN-13: 978-0-07-351717-9

ISBN-10: 0-07-351717-8

Editorial director: *Brent Gordon*

Executive editor: *Paul Ducham*

Developmental editor: *Kelly L. Delso*

Editorial assistant: *Alaina Grayson*

Executive marketing manager: *Rhonda Seelinger*

Project manager: *Kristin Bradley*

Production supervisor: *Debra R. Sylvester*

Coordinator freelance design: *Artemio Ortiz Jr.*

Typeface: *11/13 Bodoni*

Compositor: *Techbooks*

Printer: *Quebecor World Dubuque Inc.*

Library of Congress Cataloging-in-Publication Data

Bradley, Julia Case.

Advanced programming using Visual Basic 2005 / Julia Case Bradley, Anita C. Millspaugh.

—3rd ed.

p. cm.

Includes index.

ISBN-13: 978-0-07-351717-9 (alk. paper)

ISBN-10: 0-07-351717-8 (alk. paper)

1. Microsoft Visual BASIC. 2. Computer programming. I. Millspaugh, A. C. (Anita C.) II. Title.

QA76.73.B3B695 2007

005.2'768—dc22

2006044874

PREFACE

Visual Basic (VB) has become the most popular programming language for several reasons. VB is easy to learn, which makes it an excellent tool for understanding programming concepts. In addition, it has evolved into such a powerful and popular product that skilled Visual Basic programmers are in demand in the job market.

Visual Basic 2005, the latest version of VB, has many new features, especially in database access and Web application development. Visual Basic is designed to allow the programmer to develop applications that run under Windows and/or in a Web browser without the complexity generally associated with programming.

About This Text

This textbook is intended for use in an advanced programming course, which assumes completion of an introductory course. The text incorporates the basic concepts of programming, problem solving, programming logic, as well as the design techniques of an object-oriented language.

Appendix B contains a summary of topics normally covered in an introductory text, as a tool for review.

Approach

Chapter topics are presented in a sequence that allows the programmer to learn how to deal with a visual interface while acquiring important programming skills such as accessing and updating data in a relational database, developing applications for the Web and for mobile devices, and adding browser-based Help files to an application.

The chapters may be used in various sequences to accommodate the needs of the course, as well as a shorter quarter system or a semester-long course.

Changes in This Edition

This edition is a complete rewrite of the previous edition. It presents material in a sequence designed for teaching students and does not attempt to cover all topics for certification exams.

Many topics from the introductory course are presented in greater detail and demand more from the students. Many other advanced topics are presented, including displaying and updating relational databases, Web services, data structures, user controls, Help files, and mobile application development for Windows devices and for mobile browsers.

Features of This Text

Hands-On Programming Examples

These complete programming exercises guide students through the process of planning, writing, and executing Visual Basic programs.

Your Hands-On Programming Example

Create a multiple-tier project to obtain data from the employee table in the Pulse database using a stored procedure. Use an ORDER BY clause in the stored procedure to sort the data in order by last name. Display the full employee names (last name, first name, and middle initial) in alphabetic order in a combo box. Do not display the first and last name fields other than in the combo box. Show the remaining fields in text boxes for the selected record, with meaningful identifying labels. Do not include a navigation toolbar. Include a File / Exit menu item to terminate the program.

Planning the Project

Sketch a form (Figure 3.25) that your users sign off as meeting their needs.

Figure 3.25

A planning sketch for the hands-on programming example.

Plan the Objects, Properties, and Methods Plan the two tiers. Determine the objects and property settings for the form and its controls and for the data tier component. Figure 3.26 shows a diagram of the components in the two tiers.

Presentation Tier

Object	Property	Setting
EmployeeForm	Name	EmployeeForm
	Text	Employee
FullNameComboBox	Name	FullNameComboBox
Text boxes for all database fields (including Hire Date)		

CHAPTER

4

Windows Database Using Related Tables

at the completion of this chapter, you will be able to . . .

1. Explain the types of table relationships.
2. Display master-detail records.
3. Display a field from a second table using a lookup.
4. Create a search using a parameterized query and write a filter to retrieve specific data.
5. Assign data values to unbound controls.
6. Retrieve and display the parent row for a selected child row.
7. Retrieve and display an array of child rows for a selected parent row.
8. Create an application that displays data from a many-to-many relationship.
9. Select the correct locations for handling and formatting data in a multitier application.

Learning Objectives

Specific objectives tell students what will be covered in the chapter and what they will be able to do after completing the chapter.

Feedback 3.1

1. Where do you place the binding source and TableAdapter for a multi-tier project?
2. What return type is necessary for a function in the class that fills a DataSet from a TableAdapter?
3. Write the code to bind a first name label to a DataSet called CustomersDataSet. The form uses customersBindingSource. Display the FirstName field from the Customer table.

Feedback Questions

Feedback Questions encourage students to reflect on the topics covered and evaluate their understanding of details relating to that topic.

TIP

Add Imports System.Data before the Class statement; then you can omit the "Data." namespace on all declarations for data objects in code. ■

TIPs

Tips found in the margins throughout the text, help students avoid potential trouble spots in their programs and encourage them to develop good programming habits.

Case Studies

Case Studies provide continuing-theme exercises that may be used throughout the course.

New to VB Software

Coverage in this edition that is new to the Visual Basic software is highlighted with a New to VB 2005 icon.

48 ADVANCED PROGRAMMING USING VISUAL BASIC 2005 Visual Studio

Case Studies

Clayton's Cottages

Create a project for Clayton's Cottages, a small bed and breakfast. Use an MDI form with a menu, a context menu, a toolbar, and a status bar.

The About form should not be a child form. Use the My object to display the description, title, version number, and copyright attributes.

Include a splash screen with information about your program.

Menus

File	Edit	Window	Help
Exit	Guests	Title Horizontal	About
	Rooms	Title Vertical	
	Reservations	Cascade	

Set up the menus so that the open child forms display on the Window menu.

Toolbar

Include three buttons to open each of the child forms. Place ToolTips on each button.

Guests
Rooms
Reservations

Status Bar

Include the date and the time of day at the right side of the status bar. Leave a panel for text messages to the left side.

Hint: Add three labels and set the Spring property of the first label to Yes.

Context Menu

Create a context menu on the parent form that has options to display the Guests, Rooms, and Reservations forms.

Standards

- Follow naming standards for all variables, objects, and procedures.
- Menu items and controls must have keyboard access. Use standard selections when appropriate.
- Set the parent form's Icon property to an appropriate icon.

Note: The parent form icon displays in the task bar when the application is minimized.

Christian's Car Rentals

Create a project for Christian's Car Rentals. The project should contain an MDI main form with a menu, a context menu, a toolbar, and a status bar.

The About form should be a child form and contain at least your name and the date. Use the My object to display the description, title, version number, and copyright attributes.

Create child forms for each option (Customers, Vehicles, and Rentals) that simply have the title bar text indicating the form purpose and a Close button to return to the main form.

Note: These forms will be modified in later chapters.

BindingSource Objects

A **BindingSource** object facilitates binding the controls on a Windows form to the data source. The BindingSource keeps all of the form's controls displaying data from the same record (called *currency*), as well as handling the navigation, sorting, filtering, and updating of the data. In Chapter 5, you will work with methods of the BindingSource object for updating your data.

XML Data

XML is an industry-standard format for storing and transferring data. You can find the specifications for XML at <http://www.w3.org/XML>, which is the site for the World Wide Web Consortium (W3C). Although you don't need to know any XML to write database applications in VB, a few facts about XML can help you understand what is happening in your programs.

Most proprietary database formats store data in binary, which cannot be accessed by other systems or pass through Internet firewalls. Data stored in XML is all text, identified by tags, similar to HTML tags. An XML file can be edited by any text editor program, such as Notepad.

If you have seen or written any HTML, you know that opening and closing tags define elements and attributes. For example, any text between `` and `` is rendered in bold by the browser.

```
<b>This text is bold.</b><i>This is italic.</i>
```

The tags in XML are not predefined as they are in HTML. The tags can identify fields by name. For example, following are the first two records of a DataSet called AuthorsDataSet (refer to Figure 3.1), based on the Authors table in the PubS SQL Server database, represented in XML. (Later in this chapter, you will use the PubS database for VB projects.)

```
<?xml version="1.0" standalone="yes"?>
<AuthorsDataSet xmlns="AuthorsDataSet.xsd">
  <Authors>
    <au_id>172-32-1176</au_id>
    <au_fname>White</au_fname>
    <au_lname>Johnson</au_lname>
    <phone>408 496-7223</phone>
    <address>10932 Bigge Rd.</address>
    <city>Menlo Park</city>
    <state>CA</state>
    <zip>94025</zip>
    <contract>true</contract>
  </Authors>
  <Authors>
    <au_id>213-46-8915</au_id>
    <au_fname>Green</au_fname>
    <au_lname>Marjorie</au_lname>
    <phone>415 986-7020</phone>
    <address>309 63rd St. #411</address>
    <city>Oakland</city>
    <state>CA</state>
    <zip>94618</zip>
    <contract>true</contract>
  </Authors>
</AuthorsDataSet>
```

Security Issue

Any coverage that discusses security concerns is pointed out with a Security Issue icon.

jects, the data are referred to as **managed data**. Although you *can* manage the memory of your data, it is usually better and more secure to let the runtime handle it.

Your code can be integrated with classes and methods of managed code written in other programming languages. The CLR has standards for data types that allow you to pass an instance of one of your classes to a method created in a different language. Although we will not be doing any cross-language programming in this text, you should be aware of this powerful feature. Note that it is also possible to integrate methods and components created in unmanaged code, but beware of calling unmanaged code such as C++ functions or COM components as they may introduce security risks to your program.



Major Revisions and Additional Topics in This Edition

With the exception of small, concept-demonstrating programs, all programs are written as multitier applications. The chapters are reorganized to cover Web services prior to Web applications, and the middle-tier components are written as Web services.

Coverage of ASP.NET security is greatly expanded, as is the explanation of classes and of MDI applications.

Two new margin icons flag items that are “New to Visual Basic 2005” and a “Security Issue.”

The text covers the new features of Visual Studio 2005 and some new coverage of existing features, including

- Partial classes.
- The singleton design pattern for a class that should be instantiated only once.
- Validation of user input using `TryParse`.
- Displaying of messages using an `ErrorProvider` component.
- `ClickOnce` deployment.
- Generics and generic collections.
- Web themes and master pages.
- The `My` object.
- `MenuStrips`, `ToolStrips`, and `ContextMenuStrips`.
- `SQL Server Express`, which provides the advantage of local database files for development and testing.
- `TableAdapter`, `BindingSource`, and `BindingNavigator` objects.
- Stored procedures.
- The Windows database controls: `DataGridView` and `DetailsView`.
- Master-detail applications that require no code.
- Table lookups for populating a combo box from a related table.
- Use of properties and methods of the `BindingSource` to filter, navigate, and update database tables.
- Use of named table relationships.
- Field-level and record-level validation in the event handlers for changes in columns and rows.
- Consuming of a commercial Web service.
- Use of related tables in a Web service.
- The new Login controls for Web applications.
- Display and updating of data using the `Web DataSource` objects and new data components: `GridView`, `DetailsView`, and `FormView`.
- The built-in Web server, which allows development of Web applications without the necessity of running IIS. This feature also allows development



on XP Home Edition, rather than requiring XP Professional for all Web development.

- Smart Device applications, including the new device emulators.
- Use of My.Application for changing culture settings.
- The BackgroundWorker component for executing more than one thread.

New features of the Visual Studio IDE that are covered in the text include

- The DataSources window and creation of bound controls by dragging objects from the window.
- The DataSet Designer.
- The Project Designer.
- Designer-generated class diagrams.

Chapter Organization

Each chapter begins with identifiable objectives and a brief overview. Numerous coding examples as well as hands-on projects with guidance for the coding appear throughout. Thought-provoking feedback questions give students time to reflect on the current topic and to evaluate their understanding of the details. The end-of-chapter items include key terms, a chapter review, questions, programming exercises, and two case studies. The case studies provide a continuing-theme exercise that may be used throughout the course. The in-chapter programs are included on the student CD, allowing the student to test and modify existing code. Of course, it is also important to develop a complete application from scratch to really learn the programming techniques.

Chapter 1, “Visual Studio,” discusses the features of the Visual Studio IDE and the .NET Framework. Applications demonstrate how to display data from the AssemblyInfo.vb file using the My object. MDI projects, toolbars, and status bars are reviewed.

Chapter 2, “Building Multitier Programs with Classes,” reviews object-oriented terminology, creating classes, and instantiating objects. Creating and throwing exceptions from the business tier is introduced. The ErrorProvider and TryParse are demonstrated to improve the efficiency of validation in the presentation tier.

Chapter 3, “Windows Database Applications,” explains the relationship of datasets, tables, rows, and constraints. Database applications use SQL Server databases to display information in grids, combo boxes, labels, and the new database display controls.

Chapter 4, “Windows Database Using Related Tables,” explores the types of table relationships and covers the display of related tables on Windows forms. The chapter discusses how to retrieve and display the parent row for a selected child row and also an array of child rows for a selected parent row.

Chapter 5, “Windows Database Updates,” performs updates on data in a grid, in bound controls on a form, and in unbound controls. Updating