

Microsoft Visual Studio 2010 UNLEASHED



Mike Snell Lars Powers

Microsoft® Visual Studio® 2010 UNLEASHED

SAMS 800 East 96th Street, Indianapolis, Indiana 46240 USA

Microsoft® Visual Studio® 2010 Unleashed

Copyright © 2011 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-672-33081-0

ISBN-10: 0-672-33081-4

Library of Congress Cataloging-in-Publication data:

Snell, Mike.

Microsoft Visual Studio 2010 unleashed / Mike Snell, Lars Powers.

p. cm.

ISBN 978-0-672-33081-0

1. Microsoft Visual studio. 2. Web site development. 3. Application software[md]Development. I. Powers, Lars. II. Title.

TK5105.8885.M57S654 2010 006.7[md]dc22

2010029011

Printed in the United States of America First Printing: July 2010

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson Education, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Pearson offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact:

U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales +1-317-581-3793 international@pearsontechgroup.com Editor-in-Chief Karen Gettman

Executive Editor Neil Rowe

Acquisitions Editor Brook Farling

Development Editor Mark Renfrow

Managing Editor Sandra Schroeder

Senior Project Editor Tonya Simpson

Copy Editor The Wordsmithery, LLC

Indexer Lisa Stumpf

Proofreaders Debbie Williams Leslie Joseph

Technical Editor Todd Meister

Team Coordinator Cindy Teeters

Book Designer Gary Adair

Composition Mark Shirar

Contents at a Glance

	Introduction	1
Part I	An Introduction to Visual Studio 2010	
1	A Quick Tour of Visual Studio 2010	5
2	The Visual Studio IDE	
3	The .NET Languages	
Part II	An In-Depth Look at the IDE	
4	Solutions and Projects	
5	Browsers and Explorers	
6	Introducing the Editors and Designers	
7	The .NET Community: Interacting Online	
Part III	Writing and Working with Code	
8	Working with Visual Studio's Productivity Aids	
9	Testing Code.	
10	Refactoring Code	
11	Debugging Code	
12	Deploying Code	
Part IV	Extending Visual Studio	
13	Introducing the Automation Object Model	
14	Writing Macros	
15	Writing Add-ins and Wizards	
16	Extending the Code Editor Using MEF	
Part V	Creating Enterprise Applications	
17	Creating ASP.NET Applications	
18	Building Windows Forms Applications	
19	Creating Richer, Smarter User Interfaces	
20	Creating Rich Browser Applications	
21	Working with Databases	
22	Service-Oriented Applications	

iv

23	Embedding Workflow in Your Applications	1005
24	Developing Office Business Applications	1073
25	Developing Applications in the Cloud with Windows Azure	1099
	Index	1147

Table of Contents

	Introduction	1
	Who Should Read This Book?	1
	Focusing on Visual Studio Professional	2
	Downloadable Content	
	How Is This Book Organized?	
	Part I: An Introduction to Visual Studio 2010	
	Part II: An In-Depth Look at the IDE	
	Part III: Writing and Working with Code	
	Part IV: Extending Visual Studio	
	Part V: Creating Enterprise Applications	4
	Conventions Used in This Book	4
Part I	An Introduction to Visual Studio 2010	
1	A Quick Tour of Visual Studio 2010	5
	The Visual Studio Product Line	6
	Express Editions	7
	Professional Edition	8
	Premium	9
	Ultimate	9
	MSDN	
	Related Tools	
	Languages, Frameworks, and Application Templates	
	Programming Language Choices	
	The .NET Framework(s)	
	The Many Faces of a .NET Application	
	Developing Rich(er) Clients	
	Windows (WinForms)	
	Windows Presentation Foundation (WPF)	
	Office-Based Solutions	
	Creating Web Clients	
	Building Websites with ASP.NET	
	Adding Richer Features with AJAX	
	Developing for Silverlight	
	Making the Client Choice	
	Coding for the Cloud	
	Creating a Cloud Application	

	Running a Cloud Application	
	Publishing to the Cloud	
	Working with Data	
	Design Your Data	
	Separate Data Design from Storage Schema	
	Build Occasionally Connected Applications	
	Write Connected, Service-Oriented Solutions	
	Develop an Application/Business Process	
	Create and Consume Services	
	Summary	
2	The Visual Studio IDE	47
	Installing Visual Studio	
	Choosing a Language	
	Configuring Your Development Environment Settings	
	Getting Started	
	Startup Options	
	Creating Your First Project	
	Targeting Your Environment	
	Navigating the IDE	
	The Menus	
	The Many Toolbars	
	Customizing Toolbars	63
	The Solution Explorer	65
	The Text Editors	66
	The Visual Designers	
	The Toolbox	
	The Properties Window	
	Managing the Many Windows of the IDE	73
	Pinning	
	Docking	
	Navigating IDE Windows	
	Customize Your Font	
	Summary	
3	The .NET Languages	79
	Language Primer	
	Programming Objects	
	Types, Variables, and Constants	
	Operators	
	Making Decisions and Branching Code	
	Looping	
	Working with Groups of Items	

Programming with Attributes	112
Creating and Raising Events	
Language Features	
Infer a Variable's Data Type Based on Assignment	
Create an Object and Set Its Properties with a Single	
Line of Code	120
Define a Collection and Initialize Its Values (New)	122
Create an Instance of a Nonexistent Class	122
Add Methods to Existing Classes	124
Add Business Logic to Generated Code	125
Access and Query Data Using the .NET Languages	127
Write Simple, Unnamed Functions Within Your Code	129
Split an Assembly Across Multiple Files	131
Work with XML Directly Within Your Code (VB Only)	131
Remove Unused Arguments from Event Handlers (VB Only)	133
Create an Automatically Implemented Property	
(New for VB)	133
Drop the Underscore in VB for Line Continuation (New)	134
Work with Dynamic Languages/Objects (New)	
Covariance and Contravariance (New)	
The .NET Framework	142
A Map to the .NET Framework	
Summary	144

Part II An In-Depth Look at the IDE

145
172

5	Browsers and Explorers	175
	Solution Explorer	
	Visual Cues and Icons	
	Managing Solutions	
	Managing Projects	

Class View	
Toolbar	
Search Bar	
Objects Pane	
Members Pane	
Server Explorer	
Data Connections	
Server Components	
Object Browser	
Changing the Scope	
Browsing Objects	
Document Outline	
Editing Elements	
Summary	

6	Introducing the Editors and Designers	199
	The Basics	
	The Text Editor	
	Visual Studio Designers	
	Coding with the Code Editor	
	Opening an Editor	
	Writing Code in the Code Editor	
	Anatomy of the Code Editor Window	
	Code Navigation Tools	
	Searching Documents	
	Debugging in the Text Editor	
	Printing Code	
	Using the Code Definition Window	
	Creating and Editing XML Documents and Schema	
	Inferring Schema	
	Designing XML Schemas	
	Editing XSLT Style Sheets	
	Working with Cascading Style Sheets	
	Adding Style Rules	
	Defining Style Sheet Attributes	
	Developing Windows Client Applications	
	Creating a Windows Forms Project	
	Creating a Windows Presentation	
	Foundation Project	
	Developing Web Forms	
	Designing a Web Form Application	

	Authoring WinForms Components and Controls	
	Creating a New Component or Control	
	Further Notes on Writing Component Code	
	Creating Classes with the Class Designer	
	Creating a Class Diagram	
	Adding Items to the Diagram	
	Defining Relationships Between Classes	
	Defining Methods, Properties, Fields, and Events	
	Summary	
7	The .NET Community: Interacting Online	263
	The Community Features of Visual Studio	
	The Visual Studio Start Page	
	Accessing Help	
	Manage Your Help Settings	
	Using the Visual Studio Documentation	
	MSDN Forums	
	Reporting a Bug and Other Customer Feedback Options	
	Samples	
	Discovering and Consuming Shared Content	
	Examining Shared Content Types	
	Finding the Right Content	
	Installing and Storing Shared Content	
	Participating in the Community	
	Starter Kits Versus Templates	
	Creating a Project Template	
	Creating Item Templates	
	Packaging Your Creation for Wider Distribution	
	Summary	

Part III Writing and Working with Code

8	Working with Visual Studio's Productivity Aids	317
	Basic Aids in the Text Editor	
	Change Tracking	
	Coding Problem Indicators	
	Active Hyperlinking	
	Syntax Coloring	
	Outlining and Navigation	
	Code Outlining	
	Tag Navigation	

Smart Tags and Smart Tasks	
HTML Designer	
Windows Forms Designer	
Code Editor	
IntelliSense	
Complete Word	
Quick Info	
List Members	
Parameter Info	
Organize Usings	
Code Snippets and Template Code	
Brace Matching	
Customizing IntelliSense	
The Task List	
Comment Tasks	
Shortcut Tasks	
User Tasks	
Summary	

9 Testing Code

353

Unit Testing Basics	
Creating a Test Project	
Generating Tests from Existing Code	
Writing an Actual Unit Test	
Running Your Tests	
Viewing Test Results	
Configuring Testing Options and Settings	
The Unit Testing Framework	
The TestContext Class	
The Test Attribute Classes	
Unit Test Setup and Tear Down	
The Assert Classes	
Testing Your Exceptions	
Creating Data-Bound Unit Tests	
Writing ASP.NET Hosted Unit Tests	
Defining ASP.NET Attributes	
Generating an ASP.NET Unit Test	
Configuring Your Test Project Host	
Configuring Unit Test Attributes	
Defining a Connection with the	
Properties Dialog Box	
Creating Ordered Tests	

Organizing Your Tests	
The Test View Window	
The Test List Editor	
Summary	

10 Refactoring Code

393

Visual Studio Refactoring Basics	394
Invoking the Refactoring Tools	395
Previewing Changes	398
Renaming Code	400
Accessing the Rename Operation	400
Working with the Rename Dialog Box	402
Extract Method	403
Accessing the Extract Method Refactor	404
Extracting Methods	404
Generate Method Stub	410
Extract Interface	412
Accessing the Extract Interface Refactor	412
Extracting Interfaces	413
Refactor Parameters	414
Remove Parameters	415
Reorder Parameters	416
Encapsulate Field	418
Accessing Encapsulate Field	418
The Encapsulate Field Dialog Box	418
Summary	420

11 Debugging Code

421

Debugging Basics	
The Scenario	
The Many Phases of Debugging	
Debugging the Application (Self-Checking)	
Debugging Basics Summary	
The Visual Studio Debugger	
The Debug Menu and Toolbar	
Debug Options	
Stepping In, Out, and Over Code	
Indicating When to Break into Code	
Working with Tracepoints	
(When Hit Option)	
Viewing Data in the Debugger	
Using the Edit and Continue Feature	

	Advanced Debugging Scenarios	
	Remote Debugging	
	Debugging WCF Services	
	Debugging Multithreaded Applications	
	Debugging Parallel Applications	
	Debugging a Client-Side Script	
	Debugging Crash Information (Dump Files)	
	Summary	
12	Deploying Code	483
	An Overview of Client Deployment Options	
	Introducing ClickOnce Deployments	
	Introducing Windows Installer-Based	
	Deployments	
	Publishing a Project with ClickOnce	
	Publishing a Project with the Windows Installer	
	The File System Editor	
	The Registry Editor	
	The File Types Editor	
	The User Interface Editor	
	The Custom Actions Editor	
	The Launch Conditions Editor	
	Publishing an ASP.NET Website or Application	
	Using the Web Deployment Tool	
	Using the Copy Web Site Tool	
	Summary	

Part IV Extending Visual Studio

13	Introducing the Automation Object Model	507
	An Overview of the Automation Object Model	
	Object Model Versions	
	Automation Categories	
	The DTE/DTE2 Root Object	
	Solution and Project Objects	
	Controlling Projects in a Solution	
	Accessing Code Within a Project	
	Windows	
	Referencing Windows	
	Interacting with Windows	
	Text Windows and Window Panes	
	The Tool Window Types	
	Linked Windows	

Command Bars	
Documents	
Text Documents	
Command Objects	
Executing a Command	
Mapping Key Bindings	
Debugger Objects	
Automation Events	
Summary	

14 Writing Macros

563

585

Recording a Macro	
Using the Macro Explorer	
Writing Macros with the Macro IDE	
Working with Macro Projects	
Writing a Macro	
Debugging	
Handling Events	
Invoking Macros	
Summary	

15 Writing Add-ins and Wizards

Creating Your First Add-in Project586Setting Add-in Parameters587The Structure of an Add-in595The Life Cycle of an Add-in596Reacting to Commands602Managing Add-ins604A Sample Add-in: Color Palette605Getting Started606Creating the User Control606Finishing the Connect Class611Exposing Add-in Settings613Creating a Visual Studio Wizard631Creating an Add New Item Wizard635Summary640

16	Extending the Code Editor Using MEF	641
	The Extensibility Problem	641
	Creating Dynamic Applications	642

MEF Architecture	642
MEF Principles	643
Working with MEF	643
The Visual Studio Editor and MEF	644
Editor Extension Points	645
Using the Visual Studio SDK	646
Using the Extension Manager	653
Creating Your Own MEF-Based	
Editor Extension	654
Summary	663

665 . 666

Part V **Creating Enterprise Applications**

Creating ASP.NET Applications

17

The Basics of an ASP.NET Website	
Creating a New Website or Web Application	
Controlling Project Properties and Options	
	<0 -

Controlling Project Properties and Options	680
Creating Web Pages	687
Designing Your User Interface	
Determining Page Layout and Control Positioning	696
Creating a Common Look and Feel	703
Creating a User-Configurable UI	
Working with the ASP.NET Controls	
An Overview of the ASP.NET Controls	
The ASP.NET Standard Controls	738
Validation Controls	
The Login Controls	
Site Navigation Controls	
Data Controls	
User Controls	
Creating an ASP.NET MVC Application	
Understanding ASP.NET MVC	
Creating an MVC Project	
Creating Functionality Using ASP.NET MVC	
Summary	

18	Building Windows Forms Applications		
	The Basics of Form Design		
	Considering the End User		
	Understanding the Role of UI Standards		

Planning the User Interface 773

	Creating a Form	
	The Windows Forms Application Project Type	
	Form Properties and Events	
	Adding Controls and Components	
	Control Layout and Positioning	
	Using Containers	
	Control Appearance and Behavior	
	Working with ToolStrip Controls	
	Displaying Data	
	Creating Your Own Controls	801
	Subclassing an Existing Control	801
	Designing a User Control	802
	Creating a Custom Control	
	Summary	
19	Creating Richer, Smarter User Interfaces	807
	The Windows Presentation Foundation Platform	807
	Programming Model	809
	Introducing the WPF Designer	813
	XAML and Design Panes	
	Programming with WPF	
	Layout	
	Styles and Templates	
	Data Binding	
	Routed Events	
	Building a Simple Image Viewer Application	
	Starting the Layout	
	Storing the Images	
	Binding to the Images	
	Button Event Handlers and Image Effects	
	Path Selection with a Common Dialog Box	
	Summary	
20	Creating Rich Browser Applications	851
	Building Rich, Cross-Browser Interfaces with ASP.NET Aiax	
	The ASP.NET Ajax Controls	
	Creating a Page that Supports Partial Undate	854
	The ASP.NET Ajax Library/Control Toolkit	862
	Creating Unique, Rich Experiences on Windows via the Browser	869
	Choosing Between WPF Standalone and WPF XBAP	

Creating a WPF Browser Application 871

		074
	Security Considerations	
	Deploying a WPF Browser Application	
	Catting Started with Silverlight	
	Getting started with Silverlight	
	Silverlight 4.0	
	Support	
	Summary	
21	Working with Databases	889
	Creating Tables and Relationships	
	Creating a New SQL Server Database	
	Defining Tables	
	Using the Database Diagram Designer	
	Working with SQL Statements	
	Writing a Query	
	Creating Views	
	Developing Stored Procedures	
	Creating Triggers	
	Creating User-Defined Functions	
	Using Database Projects	
	Creating a Database Project	
	The Schema View	
	Building and Deploying	
	A Note on Data Tier Applications	
	Creating Database Objects in Managed Code	
	Starting a SQL CLR Project	
	Creating a Stored Procedure in C#	
	Binding Controls to Data	
	An Introduction to Data Binding	
	Auto-Generating Bound Windows	
	Forms Controls	
	Editing Typed DataSets	
	Manually Binding Windows Forms Controls	
	Data Binding in WPF Applications	
	Data Binding with Web Controls	
	Object Relational Mapping	
	An Overview of LINQ	
	Mapping Using the O/R Designer	

22	Service-Oriented Applications	955
	Service Fundamentals	
	Why ASP.NET Web Services and WCF	
	ASP.NET Web Service Applications	
	The ASP.NET Web Service Project Template	
	Coding an ASP.NET Web Service	
	Consuming an ASP.NET Web Service	
	ASP.NET Web Service Exceptions	
	WCF Service Applications	
	The WCF Project Template	
	Creating a WCF Service	
	Configuring a WCF Service	
	Consuming a WCF Service	
	Hosting and Deploying a WCF Service	
	Summary	
23	Embedding Workflow in Your Applications	1005
	Windows Workflow Fundamentals	1006
	The Components of a Workflow	1006
	The Workflow Project Templates	
	The Workflow Designer	
	The Workflow Item Templates	1011
	The Basics of Creating a Workflow	
	Working with the Built-In Workflow Activities	
	Controlling Flow	
	Runtime and Primitives	
	Error Handling	1029
	Collection Activities	
	Supporting Transactions	
	Managing Messages	
	Creating Flowchart Workflows	1038
	Creating a Workflow Application	
	The Scenario: Travel Request	
	The Application Project Templates	
	Create the Database and Data Access Library	
	Create the Vendor Reservation Service	
	Create the Custom Activity Library	
	Design the Workflow (Travel Request)	
	Creating the Client Application (Request and	
	Approval Forms)	
	Running the Travel Request Workflow Application	
	Summary	

24	Developing Office Business Applications	1073
	An Overview of Office Extension Features	
	Office Features	
	Visual Studio Office Project Types	
	Creating an Office Add-in	
	Customizing the Ribbon	
	Customizing the Task Pane	
	Creating Outlook Form Regions	
	Creating an Office Document Extension	
	Hosting Controls	
	Creating an Actions Pane	
	Storing Data in the Data Cache	
	Implementing Your Own Smart Tags	
	Summary	
25	Developing Applications in the Cloud with Windows Azure	1099
	Azure Fundamentals	
	Azure Platform Benefits	
	Setting Up Your Development Environment	
	Azure Roles	
	The Azure Project Templates	
	Azure Storage Options	
	Hello Cloud	
	Building and Deploying an Azure Application	
	The Scenario	
	Developing the Application	
	Subscribing to the Azure Services (Get an Account)	
	Creating a Storage Account	
	Creating a Hosted Service Account	
	Preparing Your Application for Publishing	
	Publish and Deploy Your Application to the Cloud	1138
	Promoting to Production	
	Promoting to Production Next Steps	

Index

1147

Introduction

Visual Studio 2010 builds on the .NET development experience that has evolved through prior editions of the product. Those editions brought us code snippets, custom project templates, refactoring, data binding wizards, smart tags, modeling tools, automated testing tools, and project and task management. The related Framework releases included Windows Presentation Foundation (WPF) for building richer client solutions, Windows Communication Foundation (WCF) to help build more dynamic service-oriented solutions, and Windows Workflow Foundation (WF) to enable structured programming around business processes.

Visual Studio 2010 adds to the innovations. It includes a new, WPF-based code editor, support for cloud development on Azure, IDE extensions with the Managed Execution Framework (MEF), additional SharePoint development support, Silverlight programming, and much more. The .NET languages have also been extended to include dynamic language support, the new F# for functional programming, and other features for helping you write better code, faster.

The tools and languages combine to increase your productivity and success rate. This book is meant to help you unlock the many tools built into Visual Studio so that you can realize these gains.

Who Should Read This Book?

Developers who rely on Visual Studio to get work done will want to read this book. It provides great detail on the many features inside the latest version of the IDE. The book covers all of the following key topics:

- ▶ Understanding the basics of solutions, projects, editors, and designers
- ▶ Writing macros, add-ins, and wizards
- Using the new Managed Extensions Framework (MEF) to write compelling extensions to the Visual Studio editor
- ▶ Debugging with the IDE
- Refactoring code
- ▶ Sharing code with team members and the larger community
- ▶ Writing ASP.NET applications (including those based on ASP.NET MVC)
- Writing Silverlight web applications
- ► Implementing Service Oriented Architecture (SOA)-based applications, and consuming services, using the Windows Communication Foundation (WCF)
- ▶ Coding with Windows forms and with Windows Presentation Foundation (WPF)
- ▶ Working with data and databases and leveraging LINQ and Entity Framework to build data-centric applications
- Creating and hosting workflow-based applications using Windows Workflow Foundation (WF)
- ▶ Writing applications for the Azure cloud

This book is primarily a tools book. We have one primary focus: detailing and explaining the intricacies of the Visual Studio 2010 IDE to enable developers to be more productive. We do, however, cover the basics of object-oriented programming with VB and C# (see Chapter 3, "The .NET Languages"). The rest of the book is about building applications using the Visual Studio toolbox. These chapters assume you can write C# or VB code and help you optimize your productivity with Visual Studio.

Focusing on Visual Studio Professional

This book represents a departure for us from our previous versions. It is focused on Visual Studio Professional. The advanced Visual Studio SKUs that include Premium, Ultimate, Team Foundation Server, and Test Professional have grown so large they need their own book. The Professional edition has also gotten bigger with more options for building more types of applications than ever before. This was a difficult decision to make, but ultimately one that frees us to go even deeper into the core Visual Studio 2010 Professional product used by the vast majority of .NET developers all over the world. Be sure to look for addi-

tional coverage of these other Visual Studio products at the SAMS site and our blog (both listed below):

- Sams: www.samspublishing.com
- ▶ Our blog: www.visualstudiounleashed.com

Downloadable Content

Many of the chapters in the book include sample content you can download. In addition, look for content that extends the samples and discussion in the book. You can find both the downloadable content and additional content at the sites listed previously (SAMS and out blog).

How Is This Book Organized?

You can read this book cover to cover, or you can pick the chapters that apply most to your current need. We sometimes reference content across chapters, but for the most part, each chapter can stand by itself. This organization allows you to jump around and read as time (and interest) permits. There are four parts to the book; each part is described next.

Part I: An Introduction to Visual Studio 2010

The chapters in this part provide an overview of what to expect from Visual Studio 2008. Readers who are familiar only with prior versions of Visual Studio will want to review these chapters. In addition, we cover the new language enhancement for the 2010, .NET 4 versions of VB and C#.

Part II: An In-Depth Look at the IDE

This part covers the core development experience relative to Visual Studio. It provides developers with a base understanding of the rich features of their primary tool. The chapters walk through the many menus and windows that define each tool. We cover the base concepts of projects and solutions, and we explore in detail the explorers, editors, and designers.

Part III: Writing and Working with Code

Part III builds on the topics discussed in Part II by digging into the powerful productivity features of Visual Studio 2008. These chapters investigate the developer productivity aids that are present in the IDE, and discuss how to best use Visual Studio for refactoring and debugging your code.

Part IV: Extending Visual Studio

For those developers interested in customizing, automating, or extending the Visual Studio IDE, these chapters are for you. We explain the automation model and then document how to use that API to automate the IDE through macros. We also cover how you can extend the IDE's capabilities by writing your own add-ins.

Part V: Creating Enterprise Applications

Part V focuses on how to work with the IDE tools to write your applications. Each chapter provides an in-depth overview of how to use Visual Studio to help you design and develop an application. We cover writing applications using ASP.NET, web services and WCF, Windows forms, WPF, WF, and working with data and databases.

Conventions Used in This Book

The following typographic conventions are used in this book:

Code lines, commands, statements, variables, and text you see onscreen appears in a monospace typeface.

Placeholders in syntax descriptions appear in an *italic monospace* typeface. You replace the placeholder with the actual filename, parameter, or whatever element it represents.

Italics highlight technical terms when they're being defined.

A code-continuation icon is used before a line of code that is really a continuation of the preceding line. Sometimes a line of code is too long to fit as a single line on the page. If you see before a line of code, remember that it's part of the line immediately above it.

The book also contains Notes, Tips, and Cautions to help you spot important or useful information more quickly.

CHAPTER 2

The Visual Studio IDE

When you're traveling over new ground, it's often wise to consult a guide. At a minimum, a quick check of the map is in order before you set out for new adventures. The same holds true for approaching a new development tool the size and breadth of Visual Studio 2010. It is wise to familiarize yourself a bit with the tool before starting that first project off on the wrong foot.

This chapter is your quick, to-the-point guide. It serves to orient you before you set out. We cover the basics of installation; configuration; booting up the IDE; and getting to know the layout of the tool in terms of projects, menus, tools, editors, and designers. Let's get started.

Installing Visual Studio

The installation of Visual Studio 2010 remains similar to that of previous versions. The application plays host to many tools. Depending on your purchase, a subset of these items are available during install (see Chapter 1, "A Quick Tour of Visual Studio 2010," for a comparison of Visual Studio editions). If you are fortunate enough to own the Team Suite Edition, you are presented with the full set of options for installation. For those with Visual Studio Professional, however, you choose one or more development languages, determine if you want to install unit testing, decide if you need the features of Microsoft Office Development, and perhaps install SQL 2008 Express. Figure 2.1 shows the setup options selection dialog box for Visual Studio Professional.

IN THIS CHAPTER

- Installing Visual Studio 47
- ▶ Getting Started 52
- ▶ Creating Your First Project 54
- Navigating the IDE 57
- Managing the Many Windows of the IDE 73



FIGURE 2.1 Visual Studio 2010 Setup Options page.

Choosing a Language

Setting up your development machine should be relatively straightforward. We suggest that most developers keep language installs to a primary language and perhaps one backup. You might use a secondary language for viewing sample code from MSDN or similar sites. Typically, this means if your primary language is Visual Basic, you install C# as a secondary language (and vice versa). Choosing to install many languages, most of which you do not intend to use, not only takes up hard drive space, but also can clutter your environment with too many choices. Having a backup language, however, helps solve the problem of finding a great bit of code you want to learn more about only to discover you can't give it a test run because it is not available in your chosen language.

Whether or not to install the additional tools is entirely up to you (and the projects you will be working on). For example, if you intend to work with a lot of local database code, you want SQL Server 2008 Express. If your project requires you to unit test your code then you should be sure to install the Unit Testing Tools.

TIP

You might change your mind about your installation selections at a later date. In this case, you can always go back and rerun setup. Rerunning setup gives you the Add or Remove Features, Repair/Reinstall, or Uninstall options.

Configuring Your Development Environment Settings

Booting the new IDE for the first time results in a dialog box asking you to choose your environment settings. As Visual Studio becomes the central tool for so many developers, testers, architects, and even project managers, it's harder and harder to satisfy them all with a single tool. To aid in this dilemma, Microsoft has created an entire set of environment settings that are configured for the usual developer type. For instance, if you set your environment to C#, the New Project dialog box automatically highlights C# projects above other languages. Figure 2.2 shows the available environment settings options.

Before you begin using the application fo activity you engage in the most, such as v predefined collection of settings to the de development activity.	r the first time, you need to specify the type of development /sual Basic or Visual C#. This information is used to apply a svelopment environment that is designed for your
(ou can choose to use a different collecti mport and Export Settings and then choo	on of settings at any time. From the Tools menu, choose ose Reset all settings.
Thoose your default environment settin General Development Settings Project Management Settings Virual Rasic Development Settings	ngs: Description: Customizes the environment to maximize code edito
Choose your default environment settin General Development Settings Project Management Settings Visual Basic Development Settings Web Development Settings Web Development (Code Only)	ngs: Description: Customizes the environment to maximize code editor screen space and improve the visibility of commands specific to C#. Increase productivity with keyboard shortcuts that are designed to be easy to learn and use.

FIGURE 2.2 The Environment Settings options dialog box.

Managing Your Settings

Only your first use of Visual Studio launches the default settings dialog box. On subsequent visits, you go straight to the tool. However, you might consider switching your environment settings if you do a lot of switching from one language to another or if you switch roles. For example, C# developers might use the C# development settings most of the time. They might then toggle to another collection of settings when switching to VB or developing a Web-only application.

You manage your environment settings from the Tools menu's Import and Export Settings option. Figure 2.3 shows the first screen in this wizard. This screen enables you to choose to execute a settings export, import, or total reset.

TIP

If you are like most developers, you are probably particular about your environment setup. There is nothing worse than having to work on a machine that has a different IDE configuration. You can be thankful that you can use the Import and Export Settings Wizard to take your IDE settings with you.



FIGURE 2.3 The Import and Export Settings Wizard.

You can choose from several setting collections when importing. There are a few default collections, including those based on language and role (such as web developer or tester in the case of Team Systems). In addition, you can select a custom settings file. Figure 2.4 shows the import settings collection options.



FIGURE 2.4 Choosing a collection of settings for import.

Another key screen to the Import and Export Settings Wizard is the settings selection screen. On this screen, you decide which options are important for export or import. This enables you to pick and choose settings you plan to either export or import. For example, you might love the way a friend has configured her code editor in terms of font and contrasting colors, but you do not want all her other settings, such as her keyboard configurations. You can accomplish this by selecting to import only her code editor settings. Figure 2.5 provides a glimpse at the granular level to which you can manage your environment settings during import and export.

ort and Export Settings Wizard		
Choose Settings to Export		
Settings with warning icons might expose intellectu default, these settings are not selected. For more info and the settings are not selected.	ual property or other sensitive information. By formation, press F1.	
Image: All Settings Image: All Settings Image: Class View Options Image: Command Window Aliases Image: Command Bar Customization <td>Description: All settings that are available for import and export. Expand this category to see more details.</td> <td></td>	Description: All settings that are available for import and export. Expand this category to see more details.	
< Previous	Next > Finish Cancel	

FIGURE 2.5 Choosing settings to export.

TIP

A great feature of the import and export settings tool is the capability to export/import any of your IDE customizations, including fonts and colors, start page customizations, menu and keyboard customizations, and more. These items can travel with you from machine to machine, so you do not have to spend valuable development time resetting them each time you rebuild or switch machines.

By default, settings are exported to your documents folder. In Windows 7, you typically find this folder at C:\Users\[user]\Documents\Visual Studio 10\Settings. Here you can find a .vssettings XML file representing the many settings for your instance of Visual Studio. This includes CurrentSettings.vssettings and any exported settings files. The

.vssettings files can be shared among users. They can also be used to migrate settings from one PC and one IDE version to another. You do so through the Import and Export Settings Wizard.

Getting Started

When you first launch Visual Studio 2010, you are presented with the Start Page for the IDE. This page has been revamped for the 2010 version to be more useful and shows off some of the new capabilities of an IDE built on top of Windows Presentation Foundation (WPF). Figure 2.6 shows an example of the Start Page.



FIGURE 2.6 The Visual Studio 2010 Start Page.

The Start Page contains a number of useful links to get you moving quickly. Starting from the upper left, you have three primary options: connect to TFS, new project, and open project. You also can launch a recent project from the left side of the screen. Across the top are tabs for getting started, guidance and resource, and latest news. These all provide access to online content. See Chapter 7, "The .NET Community: Interacting Online," for more information on working with the Start Page.

TIP

The new Start Page gives you more control over the projects that show up in the recent projects list. You can highlight a project in the list and pin it to ensure it stays in the list. You can also easily delete projects from this list.

Startup Options

If you just don't like the Start Page or prefer to launch directly into the project you'll be spending the next few months of your life working on, you can customize what happens when the IDE boots. From the Options dialog box (Tools, Options or the Settings link on the Start Page), choose the Environment node and then the Startup leaf. Figure 2.7 shows some of the options available to you at startup.

Environment	-	At startup:
General		Show Start Page
Add-in/Macros Security		Start Page news channel:
Documents		http://go.microsoft.com/fwlink/?linkid=84795&clcid=409
Extension Manager Find and Replace Fonts and Colors Import and Export Settings International Settings	Е	Download content every: 60 minutes Customize Start Page:
Keyboard		(Default Start Page)
Keyboard Startup Task List Web Browser Projects and Solutions Source Control Text Editor		Warning: Using a custom Start Page from a source that is unknown or not trusted can expose to your computer to security threats because the page and any referenced assemblies will run under the privileges of the current user.
Debugging Database Tools	-	

FIGURE 2.7 Startup options.

From here, you can configure where to get your start page news(Start Page news channel). You set this to a valid RSS feed URL. You can also use the At Startup option to tell the environment to load the last solution, show the new or open project dialog boxes, open your browser's home page, or do nothing (show an empty environment). You can also configure how often your content is automatically refreshed from the server. Finally, you have the option here to use a custom start page. For more information on custom start pages click the Customize the Start Page link on the Welcome screen of the Start Page. N

Creating Your First Project

The next, natural step is to create your first project. You might have an existing project you want to open or you might be starting fresh. In either case, creating or opening a project quickly exposes you to some of the basic project and file management features within the IDE.

To get started, you can click the File menu or the Projects link on the start page. Assuming you are using the File menu, you see the options to create a new project or website under the New submenu. Projects are simply templates that group files for Windows, Office, Web, and similar applications A website creates a set of web files that get promoted and managed as files (and not complied code).

You might have multiple projects grouped together to form a single application. In this case, each project might be grouped under a single solution. Figure 2.8 shows an example of the New Project dialog box. Notice that a Visual C#, ASP.NET Web Application is being created along with a new solution to house the project. For more information on this, see Chapter 4, "Solutions and Projects."



FIGURE 2.8 Creating a New Project.

Targeting Your Environment

Many of us work in environments that include applications built on various versions of the .NET Framework. You might be building your new applications on .NET 4.0, but still need to support one or more .NET 2.0 applications. Of course, this becomes even more prevalent as more versions are released. You do not, however, want to have to keep

multiple versions of Visual Studio on your machine. Instead, you should be able to target the version of the Framework for which the application is written. This way you can work in a single IDE and take advantage of the latest productivity enhancements.

Visual Studio 2010 supports the ability to target a specific version of the .NET Framework for an application. This means you can use a single tool to develop against many applications built on various .NET Framework flavors. Setting the .NET Framework version of an application appropriately sets the toolbox, project types, available references, and even IntelliSense inside the IDE to be in sync with the chosen .NET Framework version. Figure 2.9 shows the New Project dialog box again; this time, the .NET Framework version selection (top-center) has been highlighted.

		NET Framework 4 + Sort by: Default	 35 (11) 	Search Installed Templates
Installed Template	s	.NET Framework 2.0		Turner Minuel Co
√ Visual C#		.NET Framework 3.0 .NET Framework 3.5	Visual C#	A project for creation an application with a
Windows		.NET Framework 4		Web user interface
Web		<more frameworks=""> Tet Application</more>	Visual C#	
Office			10-10-	
Cloud		ASP.NET Empty web Application	Visual C#	
Extensibility		ASD NET MUC 2 Empty Web Application	Minual C#	
Silverlight		CR ASPERTING 2 Employ Web Application	visual C*	
Silverlight fo	Windows Phone	ASP.NET Dynamic Data Entities Web Application	Visual C#	
Test		UT /		
WCF		ASP.NET Dynamic Data Ling to SQL Web Applicat	ion Visual C#	
Workflow				
XNA Game S	tudio 4.0	ASP.NET AJAX Server Control	Visual C#	
Other Language	5			
Conter Project Types Database Test Projects		ASP.NET AJAX Server Control Extender	Visual C#	
		ASP.NET Server Control	Visual C#	
Online Templates		-		
Name:	WebApplication	4		
Location:	c:\code\		Browse	
Solution name:	WebApplication	1		✓ Create directory for solution
				Add to source control
	_		_	

FIGURE 2.9 Creating an application to target a specific version of the .NET Framework.

After you select a Framework version, the IDE automatically adjusts the available project types, IntelliSense, reference-able libraries, and similar features. For instance, if you choose to add a reference to your project, only those libraries from the target version of the Framework are available to you in the Add Reference dialog box.

You can also decide to move your application to a different (hopefully newer) version of the .NET Framework at a later date. You can do so inside the project properties dialog box (right-click your project file inside of Solution Explorer and select Properties). Figure 2.10 shows an example. Notice the Target Framework drop-down. You can change this and the IDE then resets IntelliSense, references, your toolbox, and more to the newly selected target framework.



FIGURE 2.10 Resetting the target Framework of a Web application.

NOTE

The Framework setting is per project. Therefore, you can create a single solution that contains multiple projects and each can target a different version of the .NET Framework.

Of course, you can use Visual Studio 2010 to open an existing application built on a prior version of the .NET Framework. When doing so, you have the option of upgrading or keeping it tied to its current.NET Framework version. Figure 2.11 shows an example of the dialog box you see opening an older application with Visual Studio 2010. Note that you can choose to upgrade the target Framework version or continue to target an older version.



FIGURE 2.11 Opening an older application using Visual Studio 2010.

Many environments include developers using different versions of Visual Studio. You do have to be careful about opening these older applications inside of a newer version of

Visual Studio. While the code is not affected, the actual solution file is converted to 2010 (and thus rendered useless to prior versions). If this is your environment, we recommend creating multiple versions of the solution file: one for each version of Visual Studio in use by the team.

Navigating the IDE

After you've created your first project you should get started adding features to your application. This, of course, requires that you have some base understanding of the many components of the IDE. Figure 2.12 shows a sample website inside the IDE. Notice that the IDE layout is relatively generic: Toolbox on the left, Solution Explorer on the right, and code in the middle. You should expect a similar experience for your applications (at least until you've customized things).



FIGURE 2.12 A simple Web application in the IDE.

Getting around inside the IDE is the first step to being productive. In the following sections, we break down the many items shown in Figure 2.12; it might be useful to refer to this graphic to provide overall context as we discuss a given item.

The Menus

If you've been working with previous versions of Visual Studio, you should find the Visual Studio 2010 menu bar to be standard fare. It is very intuitive; options are where you'd expect them; and new menus appear depending on your place within the IDE, the tools

you've chosen to install, and your default programming language. For example, a Refactor menu appears when you are in the C# code editor; the Project menu shows up when you have a project open; and the File menu configures itself differently depending on the programming language you are using.

Table 2.1 lists (from left to right across the IDE) some of the more common menus, along with a description of each.

Menu	Figure	Description
File	Bit Image: Construction of the second seco	The File menu is used to create new projects and websites. From here, you can also add new items. The File menu lets you save work, work with projects under source control, and print your code.
Edit	Bill Child Image: State of the st	The Edit menu is used for managing items on your Clipboard and fixing mistakes with Undo and Redo. In addition, the Edit menu provides access to important tools such as Find and Replace and IntelliSense. The fly-out menu in the graphic shows some of the advanced options available from the Edit menu, such as Format Document, which is useful to apply your formatting settings to the code with which you are working.
View	New 17 30 projew 900 migrar Schurt 71 35 Schurt Right 36 Schurt Right 37 Gig Brenzhy 38 Schurt Right 30 Schurt Right 31 Schurt Right 31 Schurt Right 32 Schurt Right 33 Schurt Right 34 Schurt Right 35 Schurt Right 36 Schurt Right 37 Schurt Right 36 Schurt Right 37 Schurt Right 30 Schurt Right<	The View menu provides access to the multitude of windows available in Visual Studio. If you lose your way (or window) in the tool, the View menu is the best place to look to find your bearings. From here, you can access the Server Explorer, Solution Explorer, Task List, and other key windows of the IDE. The fly-out menu shows the Other Windows option—the many, many windows of Visual Studio 2010.

	TABLE 2.1	Visual	Studio	2010	Menus
--	-----------	--------	--------	------	-------

TABLE 2.1 Continued

Menu	Figure	Description		
Refactor	Refactor Image: Parameter struct Image: Parameter struct	The Refactor menu (C# only) provides access to options such as renaming code elements, extracting code from a method to a new method, and promoting local variables to parameters. See Chapter 10, "Refactoring Code," for more information on refactoring.		
Website	Website Madd New Item Ctrl+Shift+A Add Existing Item Shift+Alt+A Copy Web Site Nest Related Files Add Reference Add Web Reference Add Service Reference Set as StartUp Project Start Options Start Options	The Website menu is available only when you're working with websites (and not web applications or other project types). It provides access to add new items, add references, copy your website to a deploy- ment location, and work with project dependencies. You can also set the start page for the application and access ASP.NET configuration options for the given website.		
Project	Project % Add Class Shift+Alt+C 11 Add New Rem Ctrl-Shift+A 12 Add Existing Rem Shift+Alt+A 13 Add SP.NET Folder Image: Shift+Alt+A 14 Add ASP.NET Folder Image: Shift+Alt+A 15 Build Deployment Package Image: Shift+Alt+A 16 Build Deployment Package Image: Shift+Alt+A 17 Build Deployment Package Image: Shift+Alt+A 18 Build Deployment Package Image: Shift+Alt+A 19 Package/Publish Settings Image: Shift+Alt+A 20 Show All Files Image: Shift+Alt+Alt+A 21 Add Reference Add Service Reference 22 Set as StartUp Project Image: Shift+Alt+Alt+Alt+Alt+Alt+Alt+Alt+Alt+Alt+Al	The Project menu is similar to the Website menu but is available to both web and non-web-based projects. From here, you can add new items and references to your projects, set the startup project, and change the build order for projects in your solution. In addition, you can access the Properties for a given project. This enables you to set things such as the version of the .NET Framework you are targeting, the default name- space, and many more items.		
Build	Build F6 Bebuild Solution F6 Gebuild Solution Clean Solution Build ProjectTracker Shift+F6 Rebuild ProjectTracker Clean ProjectTracker Publish ProjectTracker Batch Build Configuration Manager Configuration Manager	The Build menu enables you to invoke the given compiler(s) for your solution. From here, you can force a build or rebuild on the entire solution or an individ- ual project within the solution. You can also access the Configuration Manager from the Build menu. This dialog box enables you to control your target build in terms of debug versus release, CPU, and so on.		
TABLE 2.1 Continued

Menu	Figure	Description
Debug	Default Image of the state	The Debug menu provides developers access to the debug commands for Visual Studio. These commands include options for starting your project inside a debugging session and attaching a new debug session to an existing, executing process. In addition, you can manage debug breakpoints from this menu. The fly-out menu shows some of the other debug windows available from this menu. For more information, see Chapter 11, "Debugging Code."
Data	Data Diagram Add Sjew Diagram New Query Jable Dettch: Database Valg Modify Connection Stored Brocedure Close Connection Infine Function Table- Table-	The Data menu enables you to manage your data sources for a given application. You can add new items to your database from this menu. For additional information on working with data, see Chapter 21, "Working with Databases."
Format	Fgrmat New Style Attach Style Sheet Attach Style Sheet Background Color Set Position Font Jutify A Eont Background Color Jutify A Eont Barders and Shading Patients and Numbering Buillets and Numbering Borders and Shading Pgsition Convert to Hyperlink Insert Bookmark Align Make Same Size Order Bemove Formatting	The Format menu enables you to manipulate the layout of controls on a window or web form with respect to one another. For example, you can select a number of controls on your form and manage the verti- cal and horizontal spacing between them.
Tools	Semi Marcin II: (Semi- Concert II: (Semi- Concert II: (Semi- Concert II: (Semi- Addres Intergent, Concert II: (Semi- Addres Intergent, Concert II: (Semi- Addres Intergent, Concert II: (Semi- Concert II: (Semi- Address II: (Semi- Concert II: (Semi- II: (Semi- (Semi- II: (Semi- (Semi- II: (Semi- (Semi	The Tools menu provides access to many of the tools that ship with Visual Studio. This includes managing Visual Studio Add-Ins and Macros that extend your environment (see fly-out menu). You can also access tools for connecting to other servers and applications, and managing your IDE settings. The items in this tool menu are covered in depth throughout the book.

TABLE 2.1 Continued

Menu	Figure	Description
Table	Table Insert Iable Insert Delete Select Modity	The Table menu (available when in design view for a web form) is used exclusively for adding and manipu- lating tables on a web form. From here, you can create a new table, insert rows into an existing table, and resize table items.
Test	Test Test Load Metadata Ele Screte New Test List Brit Test Setings Solit Test Setings Windows Test Results Test List Entrop Test Results Test Result	The Test menu enables you to manage tests in Visual Studio. For example, you can use options on this menu to create a new unit test, manage existing tests, and measure test effectiveness. You can also launch test runs from here. See Chapter 12, "Instrument and Analyze Code," for more information.
Window	Window New Window Split [loat Dock Auto Hide Hide Auto Hide All Close All Documents Reset Window Layout 1 NewProject.aspx.cs* Windows	The Window menu enables you to manage the open windows in the IDE. You can hide windows, close all open windows, and turn an existing window such as the Solution Explorer from a docked window into a tabbed document.
Help	Help Image: Status of Decumentation Ctrl + F1, C Image: Status of Decumentation Ctrl + F1, C Image: Status of Decumentation Ctrl + F1, C Image: Status of Decumentation Status of Decumentation Image: Status of Decumentation Education	The Help menu provides direct access to all the help options available from Visual Studio. The Help menu can take you to the MSDN forums, and let you report a bug, review samples, and more. Finally, you can check for updates from here and, of course, access the help documentation.

NOTE

Note that each menu screenshot in Table 2.1 was taken using the C# menu default settings. In each case, Visual Basic has an equivalent, albeit slightly different, menu. In addition, the keyboard shortcut callouts in the menu items are also those of C#. Visual Basic developers should recognize a lot of them as the same. All menus can be customized to an individual developer's preference.

The Many Toolbars

Visual Studio 2010 includes close to 30 toolbars in just the professional edition. If you use a set of commands often, there is a good chance that there is a matching toolbar to group those commands. As a result, a large percentage of the toolbars are highly specialized. For example, if you are working with the Class Designer, you use the Class Designer toolbar to manage classes or change screen magnification. Or if you are building a SQL Query, you use the Query Designer toolbar. We do not cover each of these toolbars here because they are highly specialized. Instead, we stick to a quick tour to cover the common ground.

The Standard Toolbar

The Standard toolbar is present at all times during your IDE sessions (unless, of course, you customize things or turn it off). It provides quick access to all the commands you use over and over. The standard commands are on the top left: Create New Project, Add New Item, Open, Save, and Save All. These are followed by Cut, Copy, Paste, and Undo. Figure 2.13 shows the Standard toolbar in the IDE.

FIGURE 2.13 The Standard toolbar in Visual Studio 2010.

TIP

We suggest you learn the keyboard equivalents for such standard commands as Cut, Copy, Paste, Undo, and the like. In fact, most standard toolbar items have a shortcut you should learn. You can then remove many of these toolbar icons from the toolbar to save precious screen real estate for commands that have you reaching for the mouse anyway (and have harder-to-remember shortcut keys).

Additional items worth mentioning include the two navigation commands on the toolbar. These are the icons that look like a document (or code file) with an arrow on them. One icon's arrow points left and the other's points right. These navigation buttons enable you to move backward and forward through your code and your solution. They keep track of special lines in your code or windows you have visited and provide one-click access up and down this line. This can really help your productivity.

The button to the right of the navigation commands (the one that is a green arrow) is often called the Run or Play button. This initiates a build of your project and launches

you into debug mode. Moving to the right, you see options for initiating a search within your code. This capability can be handy for quickly finding the place where you left off or the place you are looking for. To the right of this are icons for quick access to displaying one of the many windows of Visual Studio. Just as with the View menu, these icons give you quick access to the Solution Explorer, Properties window, Object Browser, Toolbox, and so on. You even have an icon for Other Windows (the icon on the far right), which gives access to even more windows.

Customizing Toolbars

In the event that the standard toolbars that ship with Visual Studio don't meet your needs, you can create custom toolbars that do. Select the Tool menu's Customize item or right-click a toolbar in the IDE and select Customize to launch the Customize dialog box as shown in Figure 2.14. From here, select which toolbars to show, indicate icon size for toolbar items, turn on and off tips and shortcut keys, and more.



FIGURE 2.14 The Customize dialog box in Visual Studio 2010.

You make customizations to the toolbar by selecting an item and choosing one of the option buttons on the right (move up, move down, delete, and so on). If things get messed up, you can use the Reset All button for a selected Toolbar to revert to the default state.

The Toolbars tab on the Customize dialog box enables you to select which toolbars are visible. This dialog box also includes the New button, which enables you to create new toolbars to group existing commands. This gives you a great deal of customization options. After you've click the New button, you name your new toolbar and use the Commands tab (see Figure 2.15) to add items to your custom toolbar.

Toolbars Comma	nds					
Choose a menu or	toolbar to rearrange:					
Menu bar:	Menu Bar					
Toolbar:	Custom 1			•		
Context menu:	Editor Context Menus			v		
Controls						
				Add Command		
				Add New Menu		
-				Delete		
	Add Command				2	2
	Action Addins Build Class Diagram Data	* E	4 63 A 12 A	dd Watch pply Code Changes ttach to Process		- [11]
	Database Diagram		A	utos		
	DSL Tools Edit File		B	reak At Address reak at Function reak In File		

FIGURE 2.15 The Customize dialog box's Commands tab.

You can also configure your keyboard shortcut combinations from the Customize dialog box. Use the Keyboard button (the bottom of Figure 2.14) to bring up the Options dialog box to the environment's keyboard options screen. Figure 2.16 shows an example. First, you find a command in the list; next, you press a shortcut key to map (or remap) a combination. Notice that if the option is already assigned a shortcut key, Visual Studio warns you before you make the reassignment.

Environment	-	Apply the following additional I	keyboard mapping scheme	£			
General		Visual C# 2005		-	Reset		
Add-in/Macros Security AutoRecover Documents		Show commands containing:					
Extension Manager	Extension Manager Build.RebuildSelection						
Find and Replace	ii:	Build.RebuildSolution					
Fonts and Colors		Build.RefreshDependencies					
Import and Export Settings	Build.RunCodeAnalysisonSelec	ction					
International Settings		*			,		
Keyboard		Shortcuts for selected common					
Startup				- 1	Remove		
Task List					HUITOTE		
Web Browser		Use new shortcut in:	Press shortcut keys:				
Projects and Solutions		Global -	Fő		Assign		
Source Control				-			
Text Editor		Shortcut currently used by:					
Database Tools		Build.BuildSolution (F6 (Global	0)				
Debuaaina							

FIGURE 2.16 Options dialog box keyboard assignments.

We recommend that you do some of your own exploration into the many toolbars (and toolbar customization options) within Visual Studio. Often their usefulness presents itself only at the right moment. For instance, if you are editing a Windows form, having the Layout toolbar available to tweak the position of controls relative to one another can be a valuable timesaver. Knowing that these toolbars are available increases the likelihood that you can benefit from their value.

The Solution Explorer

The Solution Explorer enables you to group and manage the many files that make up your application. A solution simply contains multiple projects (applications). A project groups files related to its type. For instance, you can create a website, Windows Form application, class library, console application, and more. The files inside the project containers represent your code in terms of web pages, forms, class files, XML, and other related items.

Figure 2.17 shows the Solution Explorer undocked from the IDE. Note that a single solution is open (you might only have one solution open at a time), and the solution contains a few applications (called projects). One is a Windows Presentation Foundation (WPF) application (TimeTrackingInterface); another is a website (ProjectTracker); the project at the top is a class library (AppFramework).



FIGURE 2.17 The Visual Studio 2010 Solution Explorer.

You use the Solution Explorer to navigate the many items in your project. You can access an item by first selecting it and then double-clicking it. Solution Explorer opens the given designer or editor associated with the type of file you request. For example, opening a file with the extension .cs opens the C# code editor. You can also add a new item (class, image, form) to your application from here by right-clicking a project or folder and selecting the Add menu. Finally, you also use the Solution Explorer during source control scenarios to check items in and out of the source database. For more information, the Solution Explorer is covered in depth in Chapter 4.

The Text Editors

Visual Studio 2010 has several text editors or word (code) processors. Each text editor is based on a common core that provides the basic set of functionality for each editor such as the selection margin, the capability to collapse nested items, and colorization. Each editor derives from this core and is customized to give you the editors for code (C#, VB, and so on), the XML editor, the HTML (or aspx) editor, the style sheet editor, and more.

The Code Editors

The code editor, for our money, is where the magic happens. It is here that you get down to business leveraging your favorite language to define objects and their functionality. Of course, you can write code outside the Visual Studio editor, but why would you? You can also write a novel using Notepad or do your taxes by hand. A good code editor means higher productivity, plain and simple. And Visual Studio has some of the best code editors around.

The code editor is front and center when you're writing the guts of your application. It handles indentation and whitespace to make your code clean and readable. It provides IntelliSense and statement completion to free you from having to look up (or memorize) every object library and keyword. It provides shortcut snippets to help you quickly generate common code such as property definitions. It groups code into blocks; it provides color codes for keywords and comments; it highlights errors; it shows new code relative to previously compiled code. All in all, the Visual Studio code editor does quite a bit to keep you focused, organized, and productive.

The C# Code Editor

Figure 2.18 shows the C# code editor. Some items to note include the following:

- ▶ The code is grouped into logical sections along the left side. You can use the minus signs to close a whole class, method, property, or similar group. This capability enables you to hide code you are not working on at the moment. You can also create your own custom, named regions to do the same thing.
- Code lines are numbered along the left edge of the editor. This feature can be turned on or off for different code editors in the tool.
- New code is signaled inside the section groups with a colored line. Yellow is used for new code that has yet to be saved. The highlighted line turns green after a save and disappears after you close and reopen the file. This feature enables you to track where you have made changes to code during your current session.
- ▶ The name of the open code file is listed as the code window's tab across the top. The asterisk indicates that the code has changed since the last time it was saved.

😤 ProjectTracker - Microsoft Visual Studio		
File Edit View Refactor Project Build Debug Image: Im	Window - Mixe	Help ed Platforms - 🧑 "
Projectos* ×	•	Solution Explorer - Solution 'ProjectTrac 🝷 🕂 🗙
StAppFramework.Project • Project		A 🔁 🖻 🖧
<pre>4 Lusing System.Text; 5 6 Enamespace AppFramework 7 { 8 E class Project 9 { 10 private s 11 11 cftd.Alls.Secce 11 12 cftd.Alls.Secce 13 14 cftd.Alls.Secce 14 cftd.Alls.Seccee 14 cftd.Alls.Seccee</pre>		Solution 'ProjectTracker' (3 projects) AppFramework Properties References Client.cs User.cs User.cs
12 private 13 public c iii sim 15 { ♥ Single 16 get { □ ≠ siccof 17 c t f = c iiccof		ProjectTracker
10 36 1 % SortedUctionary<> 18		
 ✓ Static 		
Ready	class Syst Represent	em.String ts text as a series of Unicode characters, INS

FIGURE 2.18 The C# code editor.

- ▶ IntelliSense is invoked as you type. You can use the arrow keys to quickly find the item in the list. Hovering over the item shows details for the given item (tip text to the right). You can press the Tab key to complete the item from IntelliSense.
- ▶ The code is highlighted in various colors. By default, keywords are navy blue, comments are green, text is black, types you create are light blue, string values are red, and so on.
- ▶ The two drop-downs at the top of the code editor enable you to navigate between the classes in the file (left-side drop-down) and methods, fields, and properties within a given class (right-side drop-down).

The Visual Basic Code Editor

The Visual Basic code editor works much the same way as the C# editor. Figure 2.19 shows similar code as in Figure 2.18, this time written using the Visual Basic code editor. Some of the differences between the editors are as listed here:

- ▶ Horizontal lines are used to separate methods and properties within the editor.
- ► The IntelliSense drop-down list is filtered into a common subset and all the possible values.
- ▶ The code navigation drop-downs at the top of the code editor enable you to navigate the entire, active object hierarchy (including events). The left-side drop-down shows namespaces, objects, and events. The right-side drop-down shows all methods for the given type, including those you have not yet overridden. The items you have implemented are highlighted as bold within the list.

ProjectTracker - Microsoft Visual Studio			
File Edit View Project Build Debug	Data Tools Test Window	Help	
🛅 • 🕮 • 🧉 🖨 🖉 🔏 🖦 🥙	• 🗠 - 💭 • 🖓 🕨 Debug	- Mixe	ed Platforms 🔹 🧐 ''
12 % % * 注注日間日早	G 😂 😂 🚓 🗛 😡		
Particular and			
Rep die	(IIII (Developed)		
1\$Profile	(Declarations)		
1 Public Class Profile 2 3 Private _address As s 4 5 Private _name As Str	etn P ^{ol} IStructuralComparable		AppFramework DomainObjects My Project My Project My Project My Project Profile.vb
6 Public Property Name 7 Get 8 Return _name 9 End Get 10 Set(ByVal value As 11	String Comparer StringComparer StringComparison StringSpilOptions StringSpilOptions StringSpilOptions StructuralCompaniens VBtreadStringAttribute VDStrConv StructuralCompaniens StringSpilOptions	String	9 esents text as a series of Unicode characters. Properties ~ # X
Ready	U	13	Col 26 Ch 26

FIGURE 2.19 The Visual Basic code editor.

Of course there are many more text editors inside of Visual Studio. There are other language editors (C++ and F#), XML editors, XHTML editors, and more. Each of these have similar features to the two code editors shown here. We cover many of the specifics of these additional editors throughout the rest of the book.

Editor Customizations

Nearly every aspect of the many code editors can be customized to your every whim. From our experience, it seems no two developers see their code the same way. You can use the Options dialog box (Tools, Options) to change the editor's background color or the color and font of various text elements within the editor. You can also turn on line numbering, and manage indenting (tabs) and whitespace. You can set options based on language and editor. The full list of customizations for the editors is large.

Figure 2.20 shows the Options dialog box set for Fonts and Colors. From here, you can tweak the many display items in the editor in terms of their color, font, and font size.

If you dig a little deeper in the Options dialog box, you come across the Text Editor node in the option tree. From here, you can manipulate even more settings for the text editor in general for language-specific editors. For example, you can remove the horizontal procedure separators in the Visual Basic editor or turn off the automatic reformatting of code by the editor.

Environment	 Show settings for: 			33.
General	Text Editor		-	Use Default
Add-in/Macros Security AutoRecover	Font (bold type indicates fixed-width	1 fonts):		Size:
Documents	Consolas		•	10
Extension Manager Find and Replace	ii Display items:	Item foreground		
Fonts and Colors	Plain Text	▲ Default		Custom
Import and Export Settings International Settings Keyboard	Selected Text Inactive Selected Text Indicator Margin Line Numbers	Item background	i: •	Custom
Startup Task List Web Browser Projects and Solutions	Visible White Space Bookmark Brace Matching (Highlight) Brace Matching (Rectangle) Bracknoint (Dischlert)	Bold Sample:		
Source Control Text Editor Database Tools	Breakpoint (Enabled) Breakpoint (Error)	- ij - 1	[<mark>::0</mark> 0(0x8811);
Debuaaina	•			

FIGURE 2.20 The Options dialog box set to Fonts and Colors.

One common change we see developers make is controlling how the editor automatically formats code inside the C# editor. It seems granular control of curly braces is a big deal to those who look at code all day. For instance, you might like to see all your curly braces on separate lines or you might prefer them to start on the line that starts the given code block. Alternatively, you may want a might depending on the type of code block. Fortunately, you can control all of that from the Options dialog box. Figure 2.21 shows some of these options available for formatting C# inside the editor. Notice how the option also shows an example of how the code is formatted by the editor.



FIGURE 2.21 Controlling code formatting from the Options dialog box.

The Visual Designers

Visual Designers are the canvases that you work on using the mouse to create items such as forms via drag, drop, move, resize, and the like. Visual Studio 2010 ships with many such visual designers. Together, they enable you to build the items that make up your application. Items include Windows forms, web forms, class diagrams, XML schemas, and more.

The visual designers all work in a similar way. First, they take center stage within the IDE as tabbed windows surrounded by various menus, toolbars, and panes. Second, you use the Toolbox (we discuss this in a moment) as a palette from which you place items (such as controls) onto the given designer. You then configure each item's many properties using the Properties window.

Figure 2.22 shows the WPF Form Designer in action (the middle, highlighted tab). Note that the Toolbox is on the left and the Properties window is on the bottom right. Additionally, note that this figure shows the Layout toolbar. This toolbar enables you to easily position controls relative to one another. We cover the majority of the visual designers in depth in the coming chapters. You can also get a better overview from Chapter 6, "Introducing the Editors and Designers."



FIGURE 2.22 The WPF/XAML Designer.

The Toolbox

The Visual Studio 2010 Toolbox provides access to the many controls when you're building web and windows forms. It also provides access to nearly anything that can be dragged onto one of the many designers used for creating forms, XML schemas, class diagrams, and more. As an example, if you are building a web form, the Toolbox provides the many controls, grouped for easier access, which can be added to the form. Furthermore, if you are working with a text editor, the Toolbox enables you to save clips of text for quick access.

Figure 2.23 shows the Toolbox in a standard configuration (undocked from the IDE) for building a web form. Note that the Standard group of controls is closed to highlight some additional control groups. The many controls inside this Toolbox are covered throughout the rest of the book. The bulk of the Toolbox controls are covered in Part V, "Creating Enterprise Applications."

Toolbox	* 🗆 X
Standard	
🗉 Data	
R Pointer	
MaccessDataSource	
į Chart	
🕞 DataList	
₩E» DataPager	
DetailsView	
DynamicControl	
🔁 DynamicDataManager	
EntityDataSource	
- FormView	
GridView	
🚯 LinqDataSource	
ListView	
ColjectDataSource	
Repeater	
SiteMapDataSource	
SqlDataSource	
I., XmlDataSource	
Validation	
Navigation	
Pointer	
E Menu	
333 SiteMapPath	
TreeView	
🗄 Login	
WebParts	
AJAX Extensions	
Reporting	
III HTML	
1 General	

FIGURE 2.23 The Visual Studio Toolbox configured for a web form.

TIP

You can customize the Toolbox to your liking. For example, you can add your own groups (called tabs). You can also configure the Toolbox to show more icons on the screen at a time. As you familiarize yourself with the various standard controls, you can turn off their text descriptions and simply show them as icons. To do so, right-click the control group (tab) and uncheck List View. Figure 2.24 illustrates the additional screen real estate you gain by turning off the text descriptions.

Tool	юх									*	
😑 Standard											
k	-	Ξ	ab		~		-	A;	1	abl	A
		-	A	ab	₽ ×	9				\boxtimes	۲
8 <u>=</u>	e		ahl	$\overline{\mathbf{O}}$	**	H					
	ata										
k	l ja	<u>ii</u>	a	420	1		限		ç,		в
H	I.S	(x)	h								
± V	alida	tion									
	laviga	ation									
k		***	12								
ΞL	ogin										
± V	VcbP	arts									
± A	JAX	Exten	slons								
± K	epor	ting									
± H	ITML										
± 6	iener	al									

FIGURE 2.24 The Visual Studio Toolbox configured for more screen real estate.

The Properties Window

It seems that with every new release and every new tool, programming becomes less and less about writing code and more and more about dragging, dropping, and configuring. The many tools, controls, and rich designers that free us from the repetitive code also now require our attention in the form of maintenance. This work is typically done through the manipulation of the literally hundreds of properties that work in concert to define our application. This is where the Properties window comes into play. It enables us to control the size, appearance, and behavior of our controls. Furthermore, the Properties window groups common properties into sets for easier access. Finally, the Properties window also gives us access to connecting the events for a given control to the code inside our application.

Figure 2.25 shows the Properties window (undocked from the IDE) for a web button control. Note that the window can group similar properties into sections via banded categories, such as Appearance. You can also list properties in alphabetic order by clicking the AZ icon on the Properties window toolbar. Another item worth noting is the lightning bolt icon also on the toolbar (C# only). This gives you access to the events for the given

control. From the list of events you can select an event and wire it to code in your project (or double-click it to generate an event handler).

Pro	perties		* 🗆 ×
Bu	tton1 System.Web.UI.We	bControls.Button	•
•	21 🗉 🖌 🖃		
Þ	Accessibility		~
4	Appearance		
	BackColor		
	BorderColor		
	BorderStyle	NotSet	
	BorderWidth		=
	CssClass		
Þ	Font		
	ForeColor		
	Text	Button	
4	Behavior		
	CausesValidation	True	
	ClientIDMode	Inherit	
	CommandArgument		
	CommandName		
	Enabled	True	
	EnableTheming	True	
	EnableViewState	True	
	OnClientClick		
	PostBackUrl		*
Te Th	xt e text to be shown on the	button.	

FIGURE 2.25 The Properties window in Visual Studio 2010.

Managing the Many Windows of the IDE

To round out our whirlwind tour, we thought it important to provide you guidance on customizing and managing the plethora of windows available within the IDE (lest they leave you with a postage-stamp-size window in which to write your code). To manage these windows, you really need to know only two skills: pinning and docking.

Pinning

Pinning refers to the process of making a window stick in the open position. It is called pinning in reference to the visual cue you use to perform the act: a pushpin (refer to the highlighted area of the Solution Explorer title bar in Figure 2.26). Pinning is imperative because you sometimes want full-screen real estate for writing code or designing a form. In this case, you should unpin (hide) the various extraneous windows in your IDE. Note that when a window is unpinned, a vertical tab represents the window (see the highlighted Toolbox tab in Figure 2.26). Moving the mouse near this tab results in the window unfolding for your use. After you use it, however, it goes back to its hiding spot. Alternatively, you might be working to drop controls on a form. In doing so, you might want to pin (stick open) the Toolbox window (refer to Figure 2.22).



FIGURE 2.26 Pinning windows inside the IDE.

Docking

Docking is the process of connecting windows to various sticky spots within the IDE. Typically, this means docking to the left, top, right, or bottom of the IDE. For example, the Toolbox is, by default, docked to the left side of the IDE. You might prefer to put it at the bottom of the screen, docked below the active designer. You might also want to dock the Solution Explorer to the top of the screen and then un-pin it for quick access. You can see an example of these docking options in Figure 2.27.

You can also dock windows to one another. For example, you might want to dock the Properties window below the Solution Explorer. Or you might want the Properties window to be a tab within the same window to which the Solution Explorer is docked (refer to Figure 2.22).

To help with docking, Visual Studio 2010 has provided visual cues and helpers. First, click and hold the title bar with the mouse, and then drag the window to where you want to dock it. Visual Studio displays some docking icons.

Four icons are at the edge of the IDE, one each at the left, top, right, and bottom. These icons are used for docking the window at the given edge of the IDE. Using these icons results in the window being docked across the full length (or width) of the IDE. Figure 2.28 shows each of these icons as the Properties window is being docked.

There is also an icon that shows over the top of a window to which you might want to dock. This icon is used for docking the selected window relative to another window in the IDE. For example, you might want to dock the Properties window under the Solution Explore window (as shown in Figure 2.28). You do so with the bottom icon inside this icon group.

ProjectTracker - Microsoft Visual Stu	dio	_ 0 <u>_ X</u>
File Edit View Project Build D	ebug Data Format Tools Test Window Help	
🔯 • 🖽 • 🧉 🖬 🖉 🕺 🖏 🖄	ッ・ページ・🖏 🕨 Debug 🔹 Mixed Platforms	- 🤷 "
🖏 Solution Explorer		
Main.xaml" ×		÷ 🛯
100% Pro	file	4 doi do
		ertes
	Name	
-		F
8	Address	
	Richdata	
Design 14 DEXAML		
Window Window		
Taolhay		* 1 X
B Common		A
k □ □ ⊠ 🖬 🛄 🛤 .		=
Controls		1
	5 5 5 6 8 0 9 1 11 11 11 1 1 8 4	
		-
Ready		

FIGURE 2.27 Some docking options in the IDE.

Main.xaml*	×		 ✓ Solution Explorer Solution 'Proj ▼ ₽ ▲ ● ● ● ●
	Name Address Birthdate Su Mo Tu We Th Fr Sa 26 27 26 29 30 31 1	0	Solution 'Project Tracker' (4 projects AppFramework Project Tracker Project Tracker Properties App_Data Defa Defa Spx Properties Main.xaml File Properties Build Action Page
Design			Copy to Output Do not copy Custom Tool MSBuild:Compile Custom Tool N Advanced

FIGURE 2.28 A window being docked.

Of course, you can also undock items. This is simply the process of floating windows off by themselves (outside, or on top of, the IDE). To do so, you simply grab (click with the mouse) a window by the title bar and move it off to the side of the IDE or just don't choose a docking icon.

Finally, when working with a window, you can right-click the title bar and tell Visual Studio how the window should behave. Figure 2.29 shows the available options. The down-arrow icon on the window provides access to the same features. The Float option indicates that the window floats wherever you put it, on top of the IDE. This can be useful if you find yourself moving windows about or need to use multiple monitors. You turn off this option by choosing Dock. You can also use the Dock as tabbed Document option to add a window to the center of your IDE to behave just like a designer or code editor.



FIGURE 2.29 The IDE window options.

Navigating IDE Windows

You can navigate open windows in the IDE without touching a mouse. This keeps your fingers on the keyboard and can lead to greater productivity. Visual Studio 2010 provides a couple of options here. The first is a simple window-switching hotkey. Suppose you have a number of code windows open in the IDE. To navigate forward (left to right) through them, you can use the key combination Ctrl+- (minus sign). This is for the standard development settings in the IDE; your settings might differ. To go backward (right to left), you use Ctrl+Shift+- (minus sign). This provides faster window-switching without requiring that you scroll with the mouse or search through your solution.

You can get similar results using a visual aid called the IDE Navigator. This tool is similar to the Alt+Tab feature of Windows that allows for fast application switching. To access it,

you use Ctrl+Tab (and Ctrl+Shift+Tab). You use this key combination to open the dialog box and navigate open code windows and active tool windows. Figure 2.30 shows the result. Notice that active files are cycled through on the right. You can jump between the active tools and active file lists using the right- and left-arrow keys.



FIGURE 2.30 The IDE Navigator in action.

TIP

To change the keyboard combinations assigned to the IDE navigator, select the menu option Tools, Options. Under the Environment node, select Keyboard. Here you can set keyboard shortcut keys. You should change the settings assigned to Window.NextDocumentWindowNav and Window.PreviousDocumentWindowNav.

Customize Your Font

There is a setting called Environment Font inside the Options dialog box (Tools menu) under the Environment node, Fonts and Colors. This option enables you to set the font for the entire IDE to the selection of your choice. Figure 2.31 shows selecting this option from the list.

Changing this font changes your IDE. For example, suppose you set the Environment Font to Courier New. Dialogs, menus, the Toolbox, Solution Explorer, and more change. Figure 2.32 shows the results of such a change.



FIGURE 2.31 Setting the Environment Font.



FIGURE 2.32 The IDE with a new font setting.

Summary

The whirlwind tour is over. We've covered the basics of installation, creating your first project, and the standard items you encounter when journeying out on your own. You should now be oriented to the basic set of menus, toolbars, settings, and window management inside Visual Studio. Now that you have your bearings, you can begin pushing onward.

Index

A

about box information, setting breakpoints, 588 absolute positioning, 697 Abstract, C#, 87 AccessDataSource, 747, 939 Accessibility page, ASP.NET websites, 684-685 Accessibility Validation, Build page, ASP.NET websites, 683-684 accessing code within projects, 516-519 data with .NET languages, 127-129 Encapsulate Field, 418 Extract interface, 412 Extract method refactor, 404 Rename operation, 400-402 Accordion, 864 Action, 141 Actions pane Office document extensions, 1091 Office extension features, 1075-1076 active hyperlinking, 320 Activity class, expressing workflows as, 1020-1023 Activity Designer Library, 1009 Activity Library, workflow projects, 1008 Add Connection dialog box, 891 Add New Item Wizard, 635 dialog box, creating, 635-636 Execute method, 635

executing, 637-639 template files, 636-637 .vss files and .vsdir files. 639-640 add-in automation objects, 604-605 add-in options, 588-589 add-in parameters, 587 add-in options, 588-589 describing, 588-589 finishing wizards, 590 selecting languages, 587 setting about box information, 588 add-in projects, 598-587 parameters, 587 add-in options, 588-589 describing add-ins, 588-589 finishing wizards, 590 picking application hosts, 587-588 selecting languages, 587 setting about box information, 588 add-in settings, color palette add-in, 613 Option Page UI, 613-616 registering Options page, 616-631 Add-in Wizard, 586 AddFeedback, 1118 adding business logic to generated code, 125-127 code snippets, to Visual Studio, 343-345 content, to master pages, 717 controls to forms, 234 controls to Windows Presentation Foundation projects, 244 database entities, object relational mapping, 942-944 event declarations, 578-579 features to websites with AJAX. 30-31 items to Ribbon, 1081-1083 to toolboxes, 530

items to diagrams, Class Designer, 255-257 methods to existing classes, 124-125 projects, macro projects, 571 style rules, CSS (Cascading Style Sheets), 232 tasks. 527 text editors to text documents, 550 Web Parts, to zones, 729-731 add-ins. 289 color palette. See color palette add-in commands, reacting to, 602-603 describing add-ins, 588-589 developing, 24 IDTExtensibility2, 596 OnAddInsUpdate, 597 OnBeginShutdown, 597-598 OnConnection, 598-601 **OnDisconnection**, 601 OnStartupComplete, 601-602 managing, 604 add-in automation objects. 604-605 Office add-ins. See Office add-ins structure of, 595 addresses. 986 AddToCollection, 1031 adornments editor text adornment, 649-650 editor viewport adornment, 651-653 extension points, 646 AJAX, adding richer features to websites, 30-31 AJAX Extension Controls, 31 aligning, items, ASP.NET websites, 701-703 AlwaysVisibleControlExtender, 864 anchoring, controls, forms, 782-783 appearance, forms, 777 application hosts, picking, 587-588 Application property page, 169-170

application/business processes, developing with WF (Windows Workflow), 43-44 applications Data Tier Applications, 915-916 debugging, 423-424 breaking on errors, 426-427 continuing debugging, 431-432 debugging errors, 427-429 enabling debugging on websites. 423-425 processes, 429-430 setting breakpoints, 430-431 starting in Debug mode, 425-426 stepping to find the error, 432-433 dynamic applications, creating, 642 architecture, MEF (Managed Extensibility Framework) dependency injection, 643 naming and activation, 643 structural matching, 643 AreEqual, 375 AreEqual method, 375 AreNotEqual, 375 AreNotSame, 375 AreSame, 375 arithmetic operators, 100-101 arrays, 108-109 ASP.NET. 18, 34 building websites, 26-30 content pages, creating, 715-716 Copy Web Site Tool, 498, 503-504 master pages, 714-715 adding content, 717 nesting, 718 MVC (Model-View-Controller), 29-30 skins, 718-719 creating, 720-721 style rules, creating, 708-709

style sheets, 705-707 creating, 706-708 style toolset, 706 styles, 703-713 applying, 711 building, 708 editing, 713 managing, 709-711 themes. 718-719 applying to sites or pages, 722-724 creating, 718-719 unit tests. 383 configuring test project hosts, 385 defining attributes, 383-384 generating, 385 user-configurable UI, 724-725 adding Web Parts to zones, 729-731 creating Web Part page, 727 defining zones, 727-729 enabling users to configure pages, 731-735 Web Part controls, 725-727 Web Deployment Tool, 498, 499 Deploy SQL options, 500-501 launching publish proc, 502-503 Package/Publish options, 499 ASP.NET Aiax, 851-852 Control Toolkit, 866-868 controls, 852-854 Library/Control Toolkit, 862-864 pages that support partial updates, 854 partial-page updates, 854-860 showing server progress to users, 860-861 ASP.NET Ajax Control Toolkit, 864-866 Control Toolkit, 864-866 ASP.NET Ajax ConXBAP, 869

ASP.NET Ajax Library, 862-864 ASP.NET controls, 736 data controls, 747-749 Chart control, 748-750 features, 737-738 login controls, 742-743 configuring user authentication, 744-745 site navigation controls, 746 SiteMapPath, 746-747 Standard controls, 738-740 user controls, 749 adding to web pages, 750-752 creating, 749-750 validation controls, 740-742 ASP.NET MVC application, 752-754 advantages of, 756 execution model, 754-755 functionality, 759 creating controllers, 761-765 creating models, 759-761 creating views, 765-767 defining URL routing to controllers, 767-768 running applications, 768-769 projects, 756-759 Web Form model, 756 ASP.NET MVC 2 web role, 1107 ASP.NET Reports Web Site, 668 ASP.NET web role, 1107 ASP.NET Web Services, 667, 959-960 coding, 964-971 invoking web methods, 976 viewing formal web service descriptions, 972-974 viewing web methods, 974-976 WebMethod attribute class, 971-972 WebService attribute class. 971

WebService class, 971 consuming, 977 calling web services, 980-983 service references, 977-979 viewing web references, 979-980 exceptions, 984-985 project templates, 960-961 web service files, 961-963 WCF (Windows Communication Foundation). 957-959 ASP.NET websites, 666, 667 Accessibility page, 684-685 Build page Accessibility Validation, 683-684 Build Solution Action, 683 Start Action, 682-683 Target Framework, 683 controlling project properties and options, 680 Build page, 682-683 References, 680-682 creating, 666-667 choosing default programming languages, 674-675 choosing website locations, 668-669 directories, 675-677 file systems, 669-671 files. 677-679 FTP sites. 672-673 local IIS, 670-673 remote HTTP sites, 674 selecting Visual Studio website templates, 667-668 creating web pages, 687 adding control event handlers, 695 adding controls to pages, 688-691 adding page event handlers, 694 adding pages to your site, 687-688 responding to events, 690-693

MSBuild Options, 686 Silverlight Applications, 686-687 Start Options page, 685 Debuggers group, 686 Server section, 686 Start Action, 685 user interfaces. 695-696 aligning items, 701-703 control positioning, 698-701 creating common look and feel, 703-724 page layout and control positioning, 696-697 sizing, 701-702 assemblies, splitting across multiple files, 131 AssemblyCleanup, 370 AssemblyInitialize, 370 Assert classes, 374-376 verifying collections of objects, 377 verifying strings, 377 assertions, test data, 381-382 Assign, 1029 assigning keyboard shortcut keys to macros, 582-583 assignments, operators, 99-100 association, relationships, Class Designer, 258-259 attributes ASP.NET. 383-384 code snippets, 303-304 editing Windows Forms projects, 238 programming with, 112 exception handling, 112-114 unit tests, 386 creating ordered tests, 388 defining connections with Properties dialog box, 386-387 organizing tests, 389-390 Test List Editor, 390-392 VSTemplates, 303-304

Authenticode, 308-307 auto scaling, controls, forms, 784-785 AutoCompleteExtender, 864 auto-generating bound windows forms controls, 921-922 data binding customizing data-source mappings, 926-927 mapping data sources to controls. 924-926 automation categories, automation object model. 510-511 automation events, 560 automation object model, 508 automation categories, 510-511 DTE, 511-512 DTE2. 511-512 object model versions, 508-510 Autos window, 458-459 Azure, 1099 benefits of, 1100-1101 deploying applications to stage environment. 1139-1141 developing applications, 1114 configuring web roles, 1126-1127 creating projects, 1114-1115 creating table storage data model, 1115-1120 creating web role interfaces, 1120-1125 development environment, 1101 installing SDK, 1101-1104 Hello Cloud, 1110-1113 hosted service accounts, 1133-1134 preparing applications for publishing, 1134-1137 project templates, 1105-1107 promoting to production, 1142-1144 publishing and deploying cloud services manually, 1138-1139 with Visual Studio, 1141-1142

How can we make this index more useful? Email us at indexes@samspublishing.com

roles, 1104-1105 scenarios for building applications, 1113-1114 storage accounts, 1130-1133 storage options, 1108 blob storage, 1108 queue storage, 1109 SQL Azure, 1109 table storage, 1108 subscribing to, 1127-1129

В

behaviors, 986 binding controls to data. 840-841 binding elements, 986 bindings, 986 mapping, 558-559 blob storage, Azure, 1108 bookmark folders, code navigation tools, 210-211 bookmarks, code navigation tools, 209-210 brace matching, 345-346 C#, 345-346 Break All, 443 breaking into code, 446 breaking based on conditions, 451-452 breaking on location, 453 Breakpoints window, 448-449 Breakpoints window toolbar, 448-450 function breakpoints, 447 hit counts with breakpoints, 454-455 labeling breakpoints, 450-451 managing breakpoints, 450 recognizing breakpoints, 447-448

setting breakpoint conditions, 452-453 setting breakpoint filters, 454 setting breakpoints, 446-447 breakpoint conditions, setting, 452-453 breakpoint filters, 454 breakpoints, 447 debugging text editors, 220-222 hit counts, 454-455 labeling, 450-451 managing, 450 recognizing, 447-448 setting, 430-431, 446-447 Breakpoints window, 448-449 Breakpoints window toolbar, 448-450 browser applications, WPF (Windows Presentation Foundation), 871-873 deploying, 877-881 security. 874-876 security and site navigation, 876-877 browsers Object Browser. See Object Browser output and validation, web forms, 249-250 browsing objects, Object Browser, 194 Description pane, 195 bugs, reporting with Help menu, 284-285 build automation, TFS (Team Foundation Server), 12 build events, project properties, 170 Build menu, 59 Build page, ASP.NET websites, 682-683 Accessibility Validation, 683-684 Build Solution Action, 683 Start Action, 682-683 Target Framework, 683 Build property page, 170 Build Solution Action, Build page, ASP.NET websites, 683

built-in workflow activities, 1026
Collection activities, 1030-1031
controlling flow, 1026-1028
error handling, 1029
managing messages, 1033-1034
message-based workflows, 1034-1038
Runtime and Primitives, 1028-1029
Transaction activities, 1031-1033
business logic, adding to generated code, 125-127
BusinessEntities.Customer class, 965
button event handlers, image viewer

С

applications, 841-842

C#. 15 accessing and querying data, 128 adding business logic to generated code, 126-127 adding methods to existing classes, 124-125 arithmetic operators, 100 arrays, 108-109 assignments, 100 brace matching, 345-346 Build property page, projects, 170 classes, 80 collection classes, 109-111 comparison operators, 101 concatenation operators, 101 constructors, 84, 85 creating automatically implemented properties, 133 creating custom dynamic objects, 136 creating events, 114-115 creating instances of nonexistent classes, 123

creating objects and setting properties with a single line of code, 120-122 creating structures, 92 data types, 96 declaring variables, 97 defining collections and initializing values, 122 defining constants, 99 defining interface contracts, 91 Do...While/Unit, 107 dynamic objects, 138-139 enumerations, 86 events, handling and subscribing to, 116-117 exception handling, 113 fields and properties, 81, 82 For...Each. 106 For...Next. 105 If...Then...Else, 103 inferring variables' data type based on assignment, 118-120 inheritance, 87 logical and conditional operators, 102 methods, 82-83 organizing code with namespaces, 93, 95 overloading members, 90 overriding behavior, 88 programming with attributes, 112 project definition files, 161-163 raising events, 116 Reference Paths property page, 172 Select...Case, 104 static members and objects, 86 stored procedures, 917-920 Tuples, 111 type conversion, 98, 99 variance, 141 writing simple, unnamed functions, 129

C# code editor, 66-67 C++, 15 project definition files, 163-167 CalendarExtender, 865 call hierarchy, code navigation tools, 210-211 calling web services, 980-983 CancellationScope, 1033 Canvas control, WPF (Windows Presentation Foundation), 818-819 CascadingDropDown, 865 catalogs, MEF (Managed Extensibility Framework), 644 CatalogZone, 727 CD deployment, 484 cell edits, customizing, 929-931 cell types, displaying in forms, 801 CGI web role, 1107 change tracking, 319-320 ChangePassword, 743 changes, previewing (refactoring), 398-400 channels, 986 Chart control, 748-750 check constraints, Table Designer, 893-894 choosing languages, 48 Class Designer, 254 adding items to diagrams, 255-257 class diagrams, 254-255 displaying members, 255 defining methods, properties, fields and events, 259-261 defining relationships between classes, 257 association, 258-259 inheritance, 257 interfaces. 258 refactoring, 397-398 class diagrams, Class Designer, 254-255 displaving members, 255

Class View window, 182 Members pane, 185-187 Objects pane, 184-185 search bar. 183-184 toolbars, 182-183 ClassCleanup, 370 classes, adding methods to existing classes, 124-125 classification formats, extension points, 645 classification types, extension points, 645 ClassInitialize, 370 ClearCollection, 1031 ClickOnce, 484 publishing projects, 485-488 client applications creating pending requests pages, 1066-1068 site housekeeping, 1063 travel request pages, 1063-1065 workflow, 1007 clients choices, 33-35 developing for Silverlight, 31-34 client-side scripts, debugging, 477 cloud applications creating, 35-36 running, 36-37 cloud computing, 35 cloud services, publishing and deploying manually (Azure), 1138-1139 clouds, publishing to, 37 code accessing within projects, 516-519 extracting single lines of code, 409-410 printing, 223 colors and fonts, 223-224 renaming, 400 Rename dialog box, 402-404 Rename operation, accessing, 400-402

testing, 353-354 unit testing. See unit testing writing in code editors, 204-205 Windows Forms projects, 238-241 writing component code. WinForms. 252-254 Code Activity, expression workflows as 1018-1020 code definition window. 224-225 code editor windows, 205-206 code pane, 206-207 indicator margin, 207-208 selection margin, 208 code editors, 66-69, 203 MEF and, 644-645 editor extension points, 645-646 Extension Manager, 653-654 Visual Studio SDK. 646-653 opening, 203-204 smart tags, 328-329 writing code, 204-205 code generation properties, color palette add-in, 607-608 code navigation tools, 209 bookmark folders, 210-211 bookmarks, 209-210 call hierarchy, 210-211 line numbering, 209 code outlining, 322-324 code pane, code editor windows, 206-207 Code Snippet Inserter, 335-337 code snippets, 288, 334 adding to Visual Studio, 343-345 attributes, 303-304 Code Snippet Inserter, 335-337 creating, 338 functions, 342

surrounding code with snippets, 337-338 toolboxes, 345 XML snippet format, 339-343 Code Snippets Manager, 343 code window, refactoring via smart tags, 396-397 code with namespaces, writing simple, unnamed functions, 129-131 CodeActivity class, 1020 CodeMetricAdornment class, 657, 660-663 CodeMetricDisplayControl class, 656 coding, ASP.NET Web Services, 964-971 invoking web methods, 976 viewing form web service description, 972-974 viewing web methods, 974-976 WebMethod attribute class, 971-972 WebService attribute class, 971 WebService class, 971 coding problem indicators, 319-321 collaboration programming models, WPF (Windows Presentation Foundation), 812 TFS (Team Foundation Server), 12 CollapsiblePanelExtender, 864 Collection activities, workflows, 1030-1031 collection classes, 109-111 defining and initializing values, 122 collections, variance, 141 color palette add-in, 605-606 Connect class, 611 capturing user control events, 612-613 displaying tool window and user control, 611-612 inserting generated code, 613 exposing add-in settings, 613 Option Page UI, 613-616 registering Options page, 616-631

user controls, creating, 606 code generation properties, 607-608 handling movement over the palette. 606-607 helper routines, 608-610 signaling color selection, 610 colors, printing code, 223-224 command bars, 538-541 command members, 557 command objects, 556-557 command window, 527 executing commands, 532-533 running macros from, 583 CommandBar members, 541 commands add-ins, 602-603 executing, 558 in command window, 532-533 mapping key bindings, 558-559 CommandWindow object, 532 comment tasks. Task List. 348-350 Common dialog box, path selection, image viewer applications, 842-843 communicating with stores, 1062 communities, participating in, 291 creating project templates, 293-299 item templates, 299-300 packaging creations for distribution, 300-301 starter kits versus templates, 292 community features, 263-264 Start Page, 264-265 creating custom, 270-272 Get Started tab, 265-267 Guidance and Resources tab, 266-267 Latest News tab. 267 projects area, 264-266 selecting what to do at startup, 269-270 setting custom news channels, 268-269

CompareValidator, 742 comparison operators, 101 CompensableActivity, 1033 Compensate, 1033 Compile property page, projects, 171 Complete Word, 330-332 standard versus consume first mode, 332 complex data binding, 921 component sets, editing, 193-194 components forms, 778-780 of workflow, 1006-1008 composition container, MEF (Managed Extensibility Framework), 643-644 concatenation operators, 101 conditional operators, 101-102 **Configuration Editor, 997** configuration files, connecting data-bound unit tests. 379-381 configuration properties, solutions, 156-157 configuring breakpoints, 221-222 development environment settings, 49-52 test project hosts, 385 WCF services, 994-1000 web roles, Azure, 1126-1127 Confirm, 1033 ConfirmButtonExtender, 865 Connect class, 618-624 color palette add-in, 611 capturing user control events, 612-613 displaying tool window and user control, 611-612 inserting generated code, 613 connected applications, building, 40-42 connecting with configuration files, data-bound unit tests, 379-381 to data, data-bound unit tests, 378-379

connections, defining with Properties dialog box. 386-387 Console application, workflow projects, 1009 constants, defining, 99 consuming ASPNET Web Services, 977 calling web services, 980-983 service references, 977-979 viewing web references, 979-980 WCF services, 999-1002 container controls. 784 containers, forms, 785 FlowLayoutPanel, 786 SplitContainer, 786-787 TableLavoutPanel, 785-786 ToolStripContainer, 786-788 content, packaging with VSI. 301 with VSIX, 308-315 Content Model View, XML Schema Designer, 229-230 content pages, ASP.NET, 715-716 content types, extension points, 645 Continue command, 445 Debugging, 463-464 continuing, debugging, 431-432 contracts, 985 contravariance, 140-141 control events, Ribbon, 1083-1084 control positioning, user interfaces, ASP.NET websites, 696-701 controllers ASP.NET MVC application, 754 Functionality, 761-765 routing URLs to, 767-768 controlling flow. 1026-1028

projects Project objects, 515-516 Solution objects, 515-516 stacking behavior, 1091-1092 test settings, unit tests, 366-368 controls, 289 adding Windows Presentation Foundation projects, 244 appearance and behavior, 789 tab order, 789-790 ToolTips, 790 visual styles, 789 arranging in Windows Forms projects, 235-236 ASPNET. See ASPNET controls ASP.NET Ajax, 852-854 creating, 801 creating custom, 804-805 forms, 778-782 anchoring, 782-783 auto scaling, 784-785 docking, 783-784 inherited controls, 802 mapping data sources to, 924-926 resizing Windows Forms projects, 238 subclassing existing controls, 801-802 web forms, 245-246 Copy Web Site Tool, 498, 503-504 CorrelationScope, 1034 covariance, 140-141 crash information, debugging, 477-478 creating dump files, 478-479 with dump files, 479-481 dumping debug information, 478 Create Unit Tests Wizard, 359 CreateUserWizard, 743 CSS (Cascading Style Sheets), 231-232 style rules, adding, 232 style sheet attributes, 232

How can we make this index more useful? Email us at indexes@samspublishing.com

CSS positioning, 697 culture, end users, form design, 772-773 Custom Actions Editor. Windows Installer. 494-495 custom activity libraries creating, 1049 IsManager custom composite activity, 1050 GetUsersManagerActivity custom code activity, 1051-1052 custom comment tokens, comment tasks, Task List, 349-350 custom generic classes, 141 customer feedback options, Help menu, 284-285 customizing cell edits. 929-931 data-source mappings, 926-927 fonts, 77-78 forms, Windows Forms projects, 233-234 IntelliSense, 346-347 Ribbon, 1080-1081 adding items to, 1081-1083 control events, 1083-1084 Task pane, 1083-1085 Toolbars, 63-65 CustomValidator, 742

D

DACs, 915-916 dashboards, TFS (Team Foundation Server), 12 data, 37-38 accessing with .NET languages, 127-129 connected applications, building, 40-42 connecting to data-bound unit tests, 378-379 designing, 38

displaying in forms, 797 cell types, 801 data sources, 800-801 hierarchical data, 798 ImageList, 799-800 tabular data, 800 TreeNode Editor, 798 querying with .NET languages, 127-129 separating data design from storage schema, 39-41 visualizing, 462-464 data access custom activities. 1052-1054 data access libraries, creating, 1044-1048 data binding, 920-921 auto-generating bound windows forms controls, 921-922 customizing data-source mappings, 926-927 mapping data sources to controls. 924-926 selecting data sources, 922-924 complex data binding, 921 editing typed DataSets, 927-928 manually binding Windows Forms controls, 928 binding DataGridView, 929 binding other controls, 932 customizing cell edits, 929-931 simple data binding, 920-921 Web controls, 935 data-source controls, 938-939 GridView control, 936-937 selecting controls, 935-936 updating data with, 937-938 WPF (Windows Presentation Foundation), 828-830 WPF applications, 932-935 data caches Office extension features, 1076

storing data, Office document extensions, 1093-1094

Data Connection String property, 387 Data Connections, Server Explorer window, 187-188 data controls, ASP.NET, 747-749 Chart control, 748-750 Data menu. 60 data sources displaying in forms, 800-801 mapping to controls, 924-926 selecting, 922-924 data storage, Azure, 1100-1101 data synchronization. 42 Data Tier Applications, 915-916 data types, 96 dynamic data types. 134-135 inferring based on assignment, 118-120 Database Diagram Designer, 893-894 building table relationships, 895-897 many-to-many, 897-898 one-to-one, 897 reflexive, 897-898 creating database diagrams, 894-895 modifying table definitions, 895 database entities, adding, 942-944 database objects, creating in managed code, 917-916 creating stored procedures in C#, 917-920 starting SQL CLR projects, 916-917 database options, database projects, 912 database projects, 908-910 building and deploying, 915 database options, 912 import database schema, 912-913 project properties, 910-911 Schema View, 914 databases, workflow applications, 1043-1044

data-bound unit tests, 378 connecting to data, 378-379 connecting with configuration files, 379-381 test data in assertions. 381-382 data-driven results, running and viewing, 382-383 DataGridView, 800-801 binding, 929 DataList control, 748, 936 DataPager, 748 DataSet Designer, 927-928 DataSource, 370 data-source controls, 938-939 data-source mappings, customizing, 926-927 DataTips, 460-461 pinning, 461-462 debug information, debugging, 478 Debug menu. 60, 434 during active session, 436-439 at rest, 435-436 Debug mode, 425-426 debug options, 440-441 Debug property page, 171 Debug Source Files property page, 155 Debug toolbar, 439-440 Debug with Mixed, 480 debugger objects, 559-560 debuggers, viewing data, 457 Autos window, 458-459 DataTips, 460-461 Locals window, 457-458 pinning DataTips, 461-462 QuickWatch window, 459-460 visualizing data, 462-464 watching variables, 457

Debuggers group, Start Options page, ASP.NET websites, 686 debugging, 422 applications, 423-424 breaking on errors, 426-427 continuing debugging, 431-432 debugging errors, 427-429 enabling debugging on websites, 423-425 processes, 429-430 setting breakpoints, 430-431 starting in Debug mode, 425-426 stepping to find the error, 432-433 breaking into code, 446 breaking based on conditions, 451-452 breaking on location, 453 Breakpoints window, 448-449 Breakpoints window toolbar, 448-450 function breakpoints, 447 hit counts with breakpoints, 454-455 labeling breakpoints, 450-451 managing breakpoints, 450 recognizing breakpoints, 447-448 setting breakpoint conditions, 452-453 setting breakpoint filters, 454 setting breakpoints, 446-447 client-side scripts, 477 Continue command, 463-464 continuing, 431-432 crash information, 477-478 creating dump files, 478-479 with dump files, 479-481 dumping debug information, 478 Edit command, 463-464 enabling on websites, 423-425 ending sessions, 445-446 errors, 427-429 macros, 574-575

multithreaded applications, 466-467 breaking based on specific threads, 471 discovering and flagging threads, 467-468 inspecting threads, 469-471 managing debug processes and threads, 468-469 parallel applications, 472 Parallel Stacks Task view, 474-475 Parallel Stacks Thread view, 472-474 Parallel Stacks window, 472 Parallel Tasks window, 475-476 phases of, 422-423 processes, 429-430 remote debugging, 464-465 scenarios, 422 source file locations, solutions, 155-156 stepping through code, 441 beginning debug sessions, 441 Break All, 443 Continue, 445 ending sessions, 445-446 Run to Cursor, 441-442 Start Debugging, 442 Step Into, 443-445 Step Out, 444-445 Step Over command, 444 walking through code, 443 stored procedures, 905 text editors, 220 breakpoints, 220-222 controlling flow of running code, 222-223 tracepoints, 455-457 viewing data, Watch windows, 459 WCF services, 466 declaring, variables, 97 default constructors, 1118

Default.aspx.cs, 1121-1122 Delay, 1029 delegate variance, 141 dependency injection, MEF (Managed Extensibility Framework), 643 Deploy SQL options, Web Deployment Tool, 500-501 deploying Azure aplications cloud services manually, 1138-1139 to stage environment, 1139-1141 with Visual Studio, 1141-1142 stored procedures, 920 WCF services, 1002-1003 WPF browser applications, 877-881 deployment options ClickOnce, 484 publishing projects, 485-488 Windows Installer, 485 publishing projects, 488-490 DeploymentItem, 370 describing, add-ins, 588-589 Description pane, Object Browser, 195 design, navigation and, 28-29 design panes, WPF designer, 813-816 Property window, 815-816 zoom control, 815-816 designers, 202-203 object relational mapping, EF (Entity Framework), 949-950 designing data, 38 user controls, 802-803 adding controls, 803-804 embedding, 804-805 web forms, 245 browser output and validation. 249-250 controls, 245-246 editing markup, 246-247

formatting options, 247-248 standards compliance, 250 styles and style sheets. 248-249 tables. 247 websites, ASPNET, 27-28 workflow applications, 1052 adding workflow persistence service, 1061 communicating with stores, 1062 creating stores, 1061-1062 data access custom activities. 1052-1054 housekeeping, 1052 workflow, 1055-1061 XML schemas, 227 schema views, 228 XML Schema Designer, 228-230 XML schema explorer, 228 details window, mapping, 951-952 DetailsView control, 748, 935 developing application/business processes, WF (Windows Workflow), 43-44 Azure applications configuring web roles, 1126-1127 creating projects, 1114-1115 creating table storage data model. 1115-1120 creating web role interfaces, 1120-1125 rich clients, 19-11 Office-based solutions, 22-26 WinForms, 19-20 WPF (Windows Presentation Foundation), 20-22 for Silverlight, 31-34 stored procedures, 902-903 debugging, 905 SOL Editor with Ouerv/View. 903-905

How can we make this index more useful? Email us at indexes@samspublishing.com

development environment, Azure, 1101 installing SDK, 1101-1104 development environment settings, configuring, 49-52 development platforms, Azure, 1100 directories. ASP.NET websites. 675-677 **DISCO** (Discovery Document), 960 discovering threads, 467-468 displaving data in forms, 797 cell types, 801 data sources, 800-801 ImageList, 799-800 tabular data, 800 TreeNode Editor, 798 members, Class Designer, 255 displaying tool window and user control, data in forms, hierarchical data, 798 Do...While/Unit. 106-107 docking, controls, forms, 783-784 DockPanel control, WPF (Windows Presentation Foundation), 819-821 document members, 543 Document Outline, 195-196 editing elements, 197 Document Outline window, 326-327 Document property, 522 documents, 542-543 developing, 24 searching, 211-212 building search folder sets, 216 Find In Files/Replace In Files, 215-216 Find Results window, 216-217 Find Symbol. 217-219 finding search results, 214 fine-tuning searches, 213-214 Incremental Search, 219-220 Quick Find/Quick Replace, 212-213 Replace In Files, 217 replacing text, 214-215

text documents, 543-545 adding text, 550 editing, 545-550 editing text, 550-551 repositioning EditPoint, 551-556 DoWhile, 1027 drop handlers, extension points, 646 dropping underscores in VB line continuation, 134 DTE. 511-512 DTE2. 511-512 dump files creating, 478-479 debugging, 477-478 dumping debug information, 478 debugging with, 479-481 dump summary page, 480 dumping debug information, 478 dynamic applications, creating, 642 Dynamic Data Entities Web Site, 668 Dynamic Data Ling to SOL website, 668 dynamic data types, 134-135 dynamic languages, 134 custom dynamic objects, creating, 135-138 dynamic data types, 134-135 dynamic objects, 138-140 dynamic objects. 138-140 creating custom, 135-138 dynamic workflow, 1023-1025

Ε

Edit command, debugging, 463-464 Edit menu, 58 editing

attributes, Windows Forms projects, 238 custom component sets. 193-194 elements. Document Outline, 197 entity data model, 949 markup, web forms, 246-247 StatusStrip items, 796-797 styles, ASP.NET, 713 text documents, 545-551 typed DataSets. 927-928 VSIX manifest, 311-313 XML documents, 226 XSLT style sheets, 230-231 Editor Classifier, 647-648 editor customizations. 68-69 editor extension points, MEF (Managed Extensibility Framework), 645-646 Editor Margin, 648-649 editor text adornment, 649-650 editor viewport adornment, 651-653 EditorZone, 727 EditPoint objects, 546 repositioning, 551-556 EditPoint2 members, 547-549 EDM (Entity Data Model), 39-40 EDM Wizard, 40-41 EF (Entity Framework), object relational mapping, 947-948 designers, 949-950 editing entity data model, 949 mapping details window, 951-952 model browser, 950-951 element styles, ASP.NET, 704 elements, editing with Document Outline, 197 embedding, user controls, 804-805 Empty Web Site, 667

Encapsulate Field, 394, 418 accessing, 418 Encapsulate Field dialog box, 418-419 Encapsulate Field dialog box, 418-419 end users, form design, 772 location and culture, 772-773 ending debugging sessions, 445-446 endpoints, 985-986 entity data model editing, 949 querying, 952-953 updating data, 953-954 Entity Data Model (EDM), 39-40 Entity Framework (EF), object relational mapping, 947-948 designers, 949-950 editing entity data model, 949 mapping details window, 951-952 model browser, 950-951 EntityServices.CustomerServices class, 966-969 enumerations, programming objects, 86 EnvDTE, 508 EnvDTE80, 508 EnvDTE90, 508 EnvDTE90a, 508 EnvDTE100, 508 **Environment Font**, 77-78 environments, targeting, 54-57 error handling, workflows, 1029 errors breaking on, debugging, 426-427 debugging, 427-429 stepping to find, 432-433 event declarations, adding, 578-579 event handlers adding new event declaration, 578-579 forms, 777-779
image viewer applications, 841-842 initializing event objects, 579-580 removing unused, 133 writing, 577-578 event handling, macros, 575-577 event logs, Server Explorer window, 189 event objects, initializing, 579-580 events. 114 Class Designer, 259-261 creating, 114-115 forms, 776, 777 appearance, 777 creating event handlers, 777-779 startup location, 776-777 handling, 116-117 raising, 115-116 subscribing to, 116-117 exception handling, 112-114 exceptions ASP.NET Web Services, 984-985 Testing, 377-378 Exceptions dialog box, 427 Execute method, Add New Item Wizard, 635 executing Add New Item Wizard, 637-639 commands in command window, 532-533 execution model, ASP.NET MVC application, 754-755 ExistsInCollection, 1031 ExpectedException, 370 Export Template Wizard, 294 exporting, templates, 296-297 Express Editions, 7-8 Expression Blend, 14 Expression Design, 14 Expression Encoder, 14 Expression Media, 14

Expression Studio, 14 Expression tools, 13-14 Expression Web, 14 ext_ConnectMode members, 598-599 ext DisconnectMode members, 601 extensibility, MEF (Managed Extensibility Framework), 641-642 creating dynamic applications, 642 Extensible Application Markup Language. See XAML Extension Manager, MEF (Managed Extensibility Framework), 653-654 extensions. MEF-based editor extension. creating, 654-663 Extract interface, 394, 412 accessing, 412 Extract method, 394, 403 accessing Extract method refactor, 404 extracting methods, 404-409 Extract method refactor, accessing, 404 extracting interfaces, 413-414 methods, 404-409 single lines of code, 409-410

F

F#, 15, 80 Fail, 375 Feedback.cs data class, 1116 FeedbackDataSources.cs class, 1118-1120 FeedbackServiceContext.cs class 1117 fields Class Designer, 259-261 programming objects, 81-82 File menu, 58 file storage, Azure, 1100-1101 File System Editor, Windows Installer, 490-491 file systems, ASP.NET websites, 669-671 File Types Editor, Windows Installer, 492-493 file types, solution items, 152 files, ASP.NET, 677-679 FilteredTextBoxExtender, 865 Find In Files. 215-216 Find Results window, 216-217 Find Symbol, 89, 217-219 finding search results, 214 shared content, 289-291 fine-tuning searches, 213-214 finishing, wizards add-in parameters, 590 flagging threads, 467-468 flow, controlling, 1026-1028 flow layout, 697 Flowchart activities, 1038 workflows. 1038-1040 FlowDecision, 1038 FlowLayoutPanel, 786 FlowSwitch, 1039 fonts customizing, 77-78 printing code, 223-224 For...Each. 106 For...Next, 105-106 ForEach, 1027 foreign keys, Table Designer, 893-894 form design, 771-772 end users, 772 location and culture, 772-773 UI standards, 773 user interfaces, planning, 773-774 Format menu, 60 formatting options, web forms, 247-248 forms components, 778-780 containers, 785

FlowLayoutPanel, 786 SplitContainer, 786-787 TableLavoutPanel, 785-786 ToolStripContainer, 786-788 control appearance and behavior, 789 tab order. 789 ToolTips, 790 visual styles, 789 control layout and positioning, 780-782 anchoring, 782-783 control auto scaling, 784-785 docking, 783-784 controls, 778-780 creating, 775 displaying data, 797 cell types, 801 data sources, 800-801 hierarchical data, 798 ImageList, 799-800 tabular data, 800 TreeNode Editor, 798 properties and events, 776 appearance, 777 creating event handlers, 777-779 form events, 777 startup location, 776-777 ToolStrip controls, 790-791 creating menus, 791-793 creating status bars, 795-796 creating toolbars, 793-795 editing StatusStrip items, 796-797 Windows Forms Application projects, 775 inheriting another form's appearance, 776 startup forms, 775-776 Windows Forms projects adding controls to forms, 234

arranging controls, 235-236 creating, 232-233 customizing appearance, 233-234 layout grid, 236 resizing controls and editing attributes. 238 snap lines, 237-238 toolbox, 234-235 writing code, 238-241 Forms Designers, 778-780 control layout and positioning, 780-782 controls anchoring, 782-783 auto scaling, 784-785 docking, 783-784 FormView control, 748, 935 FTP sites, ASP.NET websites, 672-673 Func. 141 function breakpoints, 447 functionality, ASP.NET MVC application, 759 creating controllers, 761-765 creating models, 759-761 creating views, 765-767 defining URL routing to controllers, 767-768 running applications, 768-769

G

Generate From Usage scenarios, 329 generated code, adding business logic to generated code, 125-127 generated test code, 359-360 generating ASP.NET unit tests, 385 method stubs, 410-412

tests from existing code, 358-359 generated test code, 359-360 managing settings, 359 generics, groups of items, 109-111 Get Started tab, Start Page, 265-267 GetFeedback, 1118 GetUsersManagerActivity custom code activity, 1051-1052 Graph View, XML Schema Designer, 229 Grid panel, WPF (Windows Presentation Foundation), 821-824 grid sizing details, image viewer applications, 835-837 GridView control, 748, 935 data binding, 936-937 updating data, 937-938 groups of items, 107 arrays, 108-109 collection classes and generics, 109-111 Guidance and Resources tab, Start Page, 266-267

Н

handling events, 116-117 Hello Cloud, Azure, 1110-1113 Help menu, 61, 272-273 customer feedback options, 284-285 managing settings, 272-273 adding help content to local collections, 274 choosing between online and local help, 273-274 updating local help, 274-275 MSDN Forums, 275-279 navigating, 280-282 starting new threads, 279-280 thread alerts, 283

reporting bugs, 284-285 samples, 285-287 Visual Studio documentation, 275-276 helper routines, color palette add-in, 608-610 hiding members, programming objects, 89 hierarchical data, displaying in forms, 798 hit counts, breakpoints, 454-455 host controls, 1089-1091 hosted service accounts. Azure, 1110-1113 hosting WCF services, 1002-1003 workflows. 1017-1018 hosting controls, Office document extensions, 1089 host controls. 1089-1091 Windows Forms controls, 1089 hosting providers, Azure, 1100 hosts, 985 HostType, 371 housekeeping, workflow applications, 1052 HoverMenuExtender, 865 HTML designer, 327 HTTP (Hypertext Transfer Protocol), 959 hyperlinking, active hyperlinking, 320

I

icons

breakpoint icons, 447-448 Members pane, 186 Objects pane, 184 scope/access signal icons, 185 Solution Explorer, 176-179 signal icons, 179-181 ICustomerProfile, 989-990

IDE

navigating, 57 menus. 57-61 Properties windows, 72-73 Solution Explorer, 65-66 text editors. 66-69 toolbars, 62-63 Toolbox, 71-72 Visual Designers, 70 resizing, 583-584 windows, managing, 76-77 IDTCommandTarget, 595 IDTExtensibility2, 595, 596 OnAddInsUpdate, 597 OnBeginShutdown, 597-598 OnConnection, 598-601 OnDisconnection, 601 OnStartupComplete, 601-602 IDTToolsOptionsPage members, 614-615 IDTWizard interface, 631-633 lf. 1027 If...Then...Else. 102-105 Ignore, 371 IIS ASP.NET websites, 670-673 WCF services, 1003 image effects, 841-842 image viewer applications, 831-832 binding to images, 840-841 button event handlers, 841-842 image effects, 841-842 layout, 832 adding image viewers, 835 adding lists of images, 833-834 adding top menu, 833-835 grid sizing details, 835-837

path selection, with Common dialog box, 842-843 storing images, 838-840 image viewers, adding, 835 ImageList, 799-800 images adding lists of, 833-834 binding to, 840-841 storing, 838-840 Import and Export Settings Wizard, 49-50 import database schema, database projects, 912-913 Inconclusive, 375 Incremental Search, 219-220 indexes, Table Designer, 893-894 indicator margin, code editor windows, 207-208 inferring variables' data type based on assignment, 118-120 inheritance programming objects, 87 relationships, Class Designer, 257 inherited controls, 802 InitializeCorrelation, 1034 initializing, event objects, 579-580 Inline function, 907 inline styles, ASP.NET, 703-704 inspecting threads, 469-471 installing SDK. 308-309 development environment (Azure), 1101-1104 shared content, 291-292 templates, 298-299 Visual Studio, 47-48 choosing languages, 48 configuring development environment settings, 49-52 instances, creating of nonexistent classes, 122-123

IntelliSense, 67, 317-318, 329-330 brace matching, 345-346 code snippets, 334 adding to Visual Studio, 343-345 Code Snippet Inserter, 335-337 creating, 338 surrounding code with snippets, 337-338 toolboxes, 345 XML snippet format, 339-343 Complete Word, 330-332 standard versus consume first mode. 332 customizing, 346-347 extension points, 646 List Members, 333-334 Organize Usings, 334 Parameter Info. 334-335 Ouick Info. 332-333 interacting with windows, 521-524 interface contracts, defining, 90-91 interfaces extracting, 413-414 relationships, Class Designer, 258 InvokeMethod, 1029 invoking macros, 580 assigning keyboard shortcuts, 582-583 triggering macros from toolbars and menus, 581-582 web methods, 976 workflows, 1017-1018 IsFalse, 375 IsInstanceOfType, 375 IsManager custom composite activity, 1050 IsNotInstanceOfType, 375 IsNotNull, 375 IsNull, 375 IsTrue, 375

item templates, 288, 299-300

workflow projects, 1011-1012

items adding to diagrams, Class Designer, 255-257 adding to toolboxes, 530

J

join types, queries, 900-901 joins, queries, 900-901 JScript 8.0, 15

K

key bindings, mapping, 558-559 keyboard shortcut keys, assigning to macros, 582-583

L

Iabeling breakpoints, 450-451Lambda expressions, 129languages, 79add-in parameters, 587choosing, 48dynamic. See dynamic languagesF#, 80Latest News tab, Start Page, 267Launch Conditions Editor, 495-498layoutimage viewer applications, 832adding image viewers, 835adding lists of images, 833-834adding top menu, 833-835grid sizing details, 835-837

WPF (Windows Presentation Foundation), 817-818 Canvas control, 818-819 DockPanel control, 819-821 Grid panel, 821-824 StackPanel control, 824-825 layout grid, Windows Forms projects, 236 libraries Activity Designer Library, 1009 Activity Library, 1008 custom activity libraries, 1049 GetUsersManagerActivity custom code activity, 1051-1052 IsManager custom composite activity, 1050 data access libraries, creating, 1044-1048 limitations, of macros, 585-586 line continuation, dropping in VB, 134 line numbering, code navigation tools, 209 line wrapping, text editors, 200-202 LineDown, 551 LineUp. 551 linked windows, 536-537 LINQ code, 944-946 objects, 946-947 overview, 940-941 List Members, 333-334 listings The Activity Version of the EvenOdd Workflow, 1022-1023, 1025 Adding and Removing Items in the Toolbox Window, 530-532 An ASP.NET Unit Test, 384 Bookmarking For Loops in a VB Document, 544-545 The BookTravel Service Method Is for Simulation Purposes, 1049

The BusinessEntities.Customer Class, 965

C# Constructor Snippet, 339-340

The Code Activity Version of the EvenOdd Workflow, 1019-1020

Code Generated by the Add-In Wizard, 590-595

The CodeMetricAdornment Class, 660-663

The CompleteTravelRequest Custom Activity, 1054

Component Designer-Generated Code, 252-253

The Connect Class, 618-624

Contents of a Visual C# Project Definition File, 161-163

The Contoso Skin File, 721

Controlling Incremental Search, 525-526

Creating a SOAP Exception, 984-985

The Customer Links Web Part, 730

Customer Profile Web Service, 969-970

The CustomerProfile.svc.cs Implementation Code, 990-991

The Default.aspx.cs Code-behind File, 1121-1122

The EditCustomer.xaml.cs Code for Loading a Customer for Edit, 982-983

The EntityServices.CustomerServices Class, 966-969

EnvironmentEvents Module, 575-577

The Even Odd Workflow Host and Client Code, 1016-1017

An Example of a Unit Test Using Bound Data, 382

An Example Test Class, 371-373

An Example Unit Test Configuration File, 380

An Example Unit Test Showing the Use of the Assert Class, 376

Executing Commands in the Command Window, 532-533

The Extracted Interface, 414

The Extractions, 408-409

The Feedback.cs Data Class, 1116

The FeedbackDataSources.cs Class, 1118-1120

The FeedbackServiceContext.cs Class, 1117

The ICustomerProfile Interface Definition, 989-990

The Image Viewer Code Behind C#, 846-849

The Image Viewer XAML Code, 844-845

Implementing a dialog box with a Grid Panel, 822-823

Inserting Comments into a Text Window, 552-556

Linking and Unlinking Tool Windows, 537

A Long Static Method, 404-405

Macro: Expand/Collapse All Solution Explorer Nodes, 572-574

The MainPage.xaml Code-Behind, 886-887

Managed Code for Updating Employee Login Values, 918-919

Marking the Customer Class as a DataContract, 992-993

The NewRequest.aspx.cs Code-behind Class, 1063-1065

The PaletteControl Class, 624-628

The PaletteControlOptionPage Class, 629-630

The PendingRequest.aspx.cs Code-behind Class, 1067-1068

Querying the CommandBar Object, 538-539

Recorder-Generated Macro Code, 566-567

Replacing a Button's Template, 827-828

Resizing the IDE, 583

A Sample of an Auto-Generated Unit Test for a Method, 360

A Sample of Auto-Generated Unit Test for a Property, 360

Sample Solution File, 149-151

A Sample Unit Test, 361

A Sample .vstemplate File, 297-298

The Save and Complete Methods Used for Working with the Request Data, 1046-1048

- A Simple Namespace and Class Implementation, 517
- A .sitemap File, 746
- The Start of a Managed Code Stored Procedure, 917
- The Static Method After the Extractions, 408
- The Structure of a .VSContent File, 302
- Toggling Task Item Completion, 528-529

The TravelRequestMsg DataContract to be passed to the workflow, 1045

- The TravelRequestQueueAddActivity Custom Activity, 1053
- T-SQL for Updating Employee Login Values, 919-920

Use the CodeActivity<TResult> to Return a Single Value from Execute, 1051

VB Macro for Querying the Windows Collection, 523-524

The ViewFeedback.aspx.cs Code-behind File, 1124-1125

- Visual C++ Project Definition File, 163-167
- The VSContent File for the Smart Client Example, 304
- Windows Forms Designer—Generated Code, 238-241

A Word Document Project with Custom Smart Tag, 1096-1098

Writing to the Output Window, 535

XAML with Visual Studio Generated Data Binding, 933

ListSearchExtender, 865

ListView, 748

Load Test Virtual User Packs, Visual Studio Agents 2010, 13

Locals window, 457-458

location

breaking on, 453 end users, form design, 772-773 locations, choosing for websites, 668-669 logical operators, 101-102 Login, 743 login controls, ASP.NET, 742-743 configuring user authentication, 744-745 LoginName, 743 LoginStatus, 743 LoginView, 743 looping, 105 Do...While/Unit, 106-107 For...Each, 106 For...Next, 105-106

Μ

Macro Explorer, 565-567 macro projects, 568-569 adding projects, 571 sharing macros, 569-570 macros, 289, 563 debugging, 574-575 event handlers adding new event declaration, 578-579 initializing event objects, 579-580 event handling, 575-577 invoking, 580 assigning keyboard shortcuts, 582-583 triggering macros from toolbars and menus, 581-582 limitations of, 585-586 packaging, 304-306 with parameters, 583-584 recording, 564-565 running from command window, 583 sharing, 569-570 triggering from toolbars and menus, 581-582

writing, 571-574 writing with Macro IDE, 567-568 Macros IDE, 567-568 managed code, creating database objects, 917-916 starting SQL CLR projects, 916-917 stored procedures in C#, 917-920 Managed Extensibility Framework. See MEF management classes. Server Explorer window. 189-190 management events, Server Explorer window, 190-191 managing add-ins, 604 add-in automation objects, 604-605 breakpoints, 450 messages, 1033-1034 message-based workflows, 1034-1038 projects with Solution Explorer, 182 solutions with Solution Explorer, 181 styles, ASP.NET, 709-711 test results. 365-366 threads, 468-469 windows, 73 docking, 74-76 IDE windows, 76-77 pinning, 73-74 many-to-many, table relationships, 897-898 mapping data sources, to controls, 924-926 details window. 951-952 key bindings, commands, 558-559 object relational mapping. See object relational mapping vsCMElement Enumeration values, 517-519 margins, extension points, 645 markup, editing, 246-247 MaskedEditExtender, 865

master pages, ASP.NET, 714-715 adding content, 717 nesting, 718 MEF (Managed Extensibility Framework), 641 architecture, 642 dependency injection, 643 naming and activation, 643 structural matching, 643 catalogs, 644 code editor and, 644-645 editor extension points, 645-646 Extension Manager, 653-654 Visual Studio SDK. 646-653 composition container, 643-644 creating MEF-based editor extensions. 654-663 extensibility, 641-642 creating dynamic applications, 642 parts, 643 member accessibility, programming objects, 83-84 Members pane, Class View window, 185-187 menus navigating IDE, 57-61 ToolStrip controls, 791-793 triggering macros, 581-582 message queues, Server Explorer window, 190 message-based workflows, 1034-1038 messages, managing, 1033-1034 message-based workflows, 1034-1038 method stubs, generating, 410-412 methods adding to existing classes, 124-125 Class Designer, 259-261 extracting, 404-409 programming objects, 82-83 SetTabPicture method, 522 Microsoft Office, 18, 35

Microsoft Solutions Framework (MSF), 11 MIME types, XBAP, 879 ModalPopupExtender, 865 model browser, EF (Entity Framework), 950-951 Model-View-Controller (MVC), 29-30 models, ASP.NET MVC application, 754 functionality, 759-761 modifying, table definitions, Database Diagram Designer, 895 mouse processors, extension points, 646 MsBuild Options, ASP.NET websites, 686 **MSDN. 10** MSDN Code Gallery, sharing content, 268-315 MSDN Forums, 275-279 navigating, 280-282 starting new threads, 279-280 thread alerts, 283 MSF (Microsoft Solutions Framework), 11 MSI. 301 multithreaded applications, debugging, 466-467 breaking based on specific threads, 471 discovering and flagging threads, 467-468 inspecting threads, 469-471 managing debug processes and threads, 468-469 MustInherit, VB, 87 MVC (Model-View-Controller), 29-30

Ν

namespaces, organizing code, 93-95 naming and activation, MEF (Managed Extensibility Framework), 643 navigating IDE, 57 menus, 57-61 Properties windows, 72-73 Solution Explorer, 65-66

text editors, 66-69 toolbars, 62-63 Toolbox. 71-72 Visual Designers, 70 MSDN Forums, 280-282 navigation design and, 28-29 tag navigation, 325 Document Outline window, 326-327 tag navigator, 325-326 nesting, master pages, ASP.NET, 718 .NET ASPINET, 960 member accessibility, 83-84 .NET Framework, 16-19, 142 data types, 96 System.Activities, 144 System.AddIn, 142 System.Collections, 142 System.ComponentModel, 142 System.Configuration, 142 System.Data, 142 System.Diagnostics, 142-143 System.Diagnostics.Contracts, 143 System.Drawing, 143 System.Dynamic, 143 System.EnterpriseServices, 143 System.Globalization, 143 System.I0, 143 System.Ling, 143 System.Media, 143 System.Messaging, 143 System.Net, 143 System.Security, 143 System.ServiceModel, 143 System.Threading, 143 System.Web, 144

System.Windows, 144 System.Workflow.Activities, 144 System.Xml, 144

.NET languages

accessing and querying data, 127-129

adding business logic to generated code, 125-127

adding methods to existing classes, 124-125

collection classes, 122

contravariance, 140-141

covariance, 140-141

creating an instance of a nonexistent class, 122-123

creating an object and setting properties with a single line of code, 120-122

creating automatically implemented properties, 133

dropping underscores in VB line continuation, 134

inferring a variable's data type based on assignment, 118-120

removing unused arguments from event handlers, 133

splitting assemblies across multiple files, 131

variance, 141

writing simple, unnamed functions, 129-131

XML, 131-132

NetCF, 18

news channels, setting custom news channels, 268-269

nonexistent classes, creating instances of, 122-123

NotInheritable, VB, 87

0

OBA (Office Business Application), 23 **Object Browser, 192** browsing objects, 194 Description pane, 195 changing scope, 192-193 editing custom component sets. 193-194 Rename operation, 400 object model versions, automation object model, 508-510 object relational mapping, 939-940 entity data model, updating data, 953-954 Entity Framework (EF), 947-948 designers, 949-950 editing entity data model, 949 mapping details window, 951-952 model browser, 950-951 LINO code, 944-946 objects, 946-947 overview. 940-941 0/R Designer, 941 adding database entities, 942-944 querying against entity data model, 952-953 ObjectDataSource, 939 objects, creating and setting properties with a single line of code, 120-122 Objects pane, Class View window, 184-185 Office, 18 Office add-ins. 1079 creating Outlook form regions, 1085-1088 customizing Task pane, 1083-1085 Ribbon, customizing, 1080-1081 Office Business Application (OBA), 23

Office document extensions, 1087-1089 Actions pane, controlling stacking behavior, 1091-1092 creating Actions pane, 1091 hosting controls, 1089 host controls, 1089-1091 Windows Forms controls, 1089 smart tags, 1095-1098 storing data in data caches, 1093-1094 Office extension features, 1074, 1075 Actions pane, 1075-1076 data caches. 1076 Ribbon, 1076-1077 smart tags, 1077 Task panes, 1075-1076 Office project types, 1077-1079 Office-based solutions, developing rich clients, 22-26 OnAddInsUpdate, IDTExtensibility2, 597 OnBeginShutdown, IDTExtensibility2, 597-598 OnConnection, IDTExtensibility2, 598-601 OnDisconnection, IDTExtensibility2, 601 **OnSizeChange**, 659 OnStartupComplete, IDTExtensibility2, 601-602 opening code editors, 203-204 operating systems, Azure, 1100 operators, 99 arithmetic, 100-101 assignments, 99-100 comparison, 101 concatenation, 101 logical and conditional, 101-102 Option Page UI, 613-616 options ASPNET websites, 680 Build page, 682-683 references, 680-682 extension points, 646

Options page, registering, 616-631 0/R Designer, object relational mapping, 941 adding database entities, 942-944 ordered tests, creating, 388 Organize Usings, 334 organizing code with namespaces, 93-95 tests. 389-390 outlining code outlining, 322-324 Outlining menu, 324-325 Outlining menu, 324-325 Outlook form regions, 1085-1088 output window, 527, 533-535 OutputWindowPane members, 535 overloading members, programming objects, 89-90 overriding behavior, programming objects, 87-89

Ρ

Package/Publish options, Web Deployment Tool, 499 packaging content with VSI, 301 with VSIX, 308-315 macros, 304-306 page layout, user interfaces, ASP.NET websites, 696-697 page-level styles, ASP.NET, 704 PaletteControl class, 624-628 PaletteControlOptionPage class, 629-630 Parallel, 1027

parallel applications, debugging, 472 Parallel Stacks Task view, 474-475 Parallel Stacks Thread view, 472-474 Parallel Stacks window, 472 Parallel Tasks window, 475-476 Parallel Stacks Task view, 474-475 Parallel Stacks Thread view, 472-474 Parallel Stacks window, 472 Parallel Tasks window, 475-476 ParallelForEach, 1027 Parameter Info, 334-335 parameters add-in parameters, 587 add-in options, 588-589 describing add-ins, 588-589 finishing wizards, 590 picking application hosts, 587-588 selecting languages, 587 setting about box information, 588 macros, 583-584 refactoring, 414-415 removing, 415-416 reordering, 416-418 templates, 294-295 partial-page updates, ASP.NET Ajax, 854-860 participating in communities, 291 creating project templates, 293-299 item templates, 299-300 packaging creations for distribution, 300-301 startker kits versus templates, 292 parts, MEF (Managed Extensibility Framework), 643 PasswordRecovery, 743 PasswordStrength, 866 path selection, image viewer applications, 842-843 pending request pages, client applications, 1066-1068

performance counters, Server Explorer window, 190 Persist activity, 1028 phases of, debugging, 422-423 Pick, 1027 pinning DataTips, 461-462 windows, 73-74 planning, user interfaces, form design, 773-774 PopupControlExtender, 866 Preview Changes dialog box, 398 previewing, changes, refactoring, 398-400 primary keys, Table Designer, 893 Primitive activities, 1028-1029 printing code, 223 colors and fonts. 223-224 processes, debugging, 429-430 productivity aids, 317-318 IntelliSense, 329-330 brace matching, 345-346 code snippets. See code snippets Complete Word, 330-332 customizing, 346-347 List Members, 333-334 Organize Usings, 334 Parameter Info, 334-335 Ouick Info. 332-333 navigation, tag navigation, 325 outlining, code outlining, 322-324 smart tags code editors, 328-329 HTML designer, 327 Windows Forms designer, 327-328 Task List, 348 comment tasks, 348-350 shortcut tasks, 349 user tasks, 350

text editors active hyperlinking, 320 change tracking, 319-320 coding problem indicators, 319-321 syntax coloring, 321-322 programming with attributes. 112 exception handling, 112-114 with Server Explorer, 191-192 programming languages, 14-16 choosing, for ASP.NET websites, 674-675 JScript 8.0, 15 Visual Basic, 15 Visual C#, 15 Visual C++. 15 Visual F#. 15 programming models, WPF (Windows Presentation Foundation), 809-811 collaboration, 812 syntax simplicity, 812 XAML (Extensible Application Markup Language), 811-812 programming objects, 80 classes, 80 constructors, 84-85 creating structures, 91-93 defining interface contracts, 90-91 enumerations, 86 fields and properties, 81-82 hiding members, 89 inheritance, 87 member accessibility, 83-84 methods, 82-83 organizing code with namespaces, 93-95 overloading members, 89-90 overriding behavior, 87-89 static members and objects, 85-86

project definition files, 161-167 project dependencies, solutions, 155 project items, 167-168 project management, TFS (Team Foundation Server), 11 Project menu, 59 Project objects, 512 controlling projects, 515-516 project properties, 168-169 Application property page, 169-170 build events, 170 Build property page, 170 Compile property page, 171 database projects, 910-911 Debug property page, 171 Publish property page, 171 Reference Paths property page, 172 References property page, 171-172 Resources property page, 172 Security property pages, 172 Settings property page, 172 Signing property page, 172 Project property, 522 project templates, 288 Azure, 1114-1115 creating, 293-299 ProjectItem property, 522 projects accessing code within, 516-519 add-in projects. See add-in projects ASP.NET MVC application, 756-759 Azure applications, 1114-1115 controlling Project objects, 515-516 Solution objects, 515-516 creating, 54, 157-160 website projects, 160-161

database projects. See database projects defined, 145 defining in Silverlight applications, 882-884 macro projects, 568-569 adding projects, 571 sharing macros, 569-570 managing with Solution Explorer, 182 project definition files, 161-167 project items, 167-168 project properties, 168-169 build events, 170 Build property page, 170 Compile property page, 171 Debug property page, 171 Publish property page, 171 Reference Paths property page, 172 References property page, 171-172 Resources property page, 172 Security property pages, 172 Settings property page, 172 Signing property page, 172 publishing with ClickOnce, 485-488 with Windows Installer, 488-490 targeting your environment, 54-57 types of, 159 WCF service projects. See WCF service projects workflow projects. See workflow projects projects area, Start Page, 264-266 properties ASP.NET websites, 680 Build page, 682-683 references, 680-682 Class Designer, 259-261 creating automatically implemented properties, VB, 133 Data Connection String property, 387 database projects, 910-911

Document, 522 forms, 776 appearance, 777 startup location, 776-777 programming objects, 81-82 Project. 522 project properties, 168-169 Application property page, 169-170 ProjectItem property, 522 solution properties, 153-154 Window.CommandBars property, 538 Properties dialog box, defining connections, 386-387 Properties windows, 72-73 Property window, WPF designer, 815-816 Publish property page, 171 Publish Wizard, 485 publishing Azure applications, 1134-1137 cloud services manually, 1138-1139 with Visual Studio, 1141-1142 to clouds, 37 projects ClickOnce, 485-488 Windows Installer, 488-490 Web Deployment Tool, 502-503

Q

queue storage, Azure, 1109
queries, writing, 898-900
 query types, 901-902
 specifying joins and join types, 900-901
Query Designer, 903-905
query types, 901-902
QueryExtender, 748

querying
data with .NET languages, 127-129
entity data model, 952-953
Quick Find, 89, 212-213
Quick Info, 332-333
Quick Replace, 212-213
QuickWatch window, 459-460

R

RAD (Rapid Application Development), 771 raising events, 115-116 RangeValidator, 742 Rapid Application Development. See RAD reacting, to commands, add-ins, 602-603 Receive, 1034 ReceiveAndSendReply, 1034 recognizing breakpoints, 447-448 recording macros, 564-565 Refactor menu, 59, 395-396 refactoring, 393-394 Class Designer, 397-398 Encapsulate Field, 418 accessing, 418 Encapsulate Field dialog box, 418-419 Extract interface, 412 accessing Extract Interface refactor, 412 Extract method, 403 accessing Extract method refactor, 404 extracting interfaces, 413-414 extracting methods, 404-409 generating a method stub, 410-412 parameters, 414-415 removing, 415-416 reordering, 416-418

previewing changes, 398-400 refactoring tools, 394 invoking, 395-394 Refactor menu. 395-396 smart tags, Code window, 396-397 references, ASP.NET websites, 680-682 References Paths property page, 172 References property page, 171-172 referencing windows, 520-521 reflexive relationships, 897-898 registering Options page, 616-631 Registry Editor, Windows Installer, 491-492 RegularExpressionValidator, 742 relationships Class Designer, 257 association. 258-259 inheritance, 257 interfaces. 258 table relationships, 895-897 many-to-many, 897-898 one-to-one. 897 reflexive. 897-898 relative positioning, 697 remote debugging, 464-465 remote HTTP sites, ASP.NET websites, 674 **Remove Parameters**, 394 **Remove Parameters operation**, 415 RemoveFromCollection, 1031 removing parameters, refactoring, 415-416 tasks. 527 unused arguments from event handlers, 133 Rename, 394 Rename dialog box, 402-404 Rename operation, accessing, 400-402 renaming code, 400 Rename dialog box, 402-404 Rename operation, accessing, 400-402

Reorder Parameters, 394 Reorder Parameters operation, 417 reordering, parameters, refactoring, 416-418 ReorderList. 866 Repeater control, 748, 936 Replace In Files, 215-217 replacing text, 214-215 reporting bugs, Help menu, 284-285 TFS (Team Foundation Server), 12 repositioning EditPoint objects, 551-556 RequiredFieldValidator, 742 requirements management, TFS (Team Foundation Server), 11 resizing controls, Windows Forms projects, 238 IDE. 583-584 Resources property page, 172 results, data-driven results, running and viewing, 382-383 Rethrow, 1029 Ribbon customizing, 1080-1081 adding items to, 1081-1083 control events, 1083-1084 Office extension features, 1076-1077 rich clients, developing, 19-11 Office-based solutions, 22-26 WinForms, 19-20 WPF (Windows Presentation Foundation), 20-22 roles, Azure, 1104-1105 routed events, WPF (Windows Presentation Foundation), 830-831 Run to Cursor, 441-442 running ASPNET MVC application, 768-769 cloud applications, 36-37

data-driven results, 382-383 macros from command window, 583 unit tests, 362 WCF services, 993-994 workflow applications, 1068-1070 Runtime activities, 1028-1029

S

samples, 289 Help menu, 285-287 Scalar-valued function, 908 scenarios, workflow applications, 1040-1041 schema explorer, XML schemas, 228 Schema View, database projects, 914 schema views, XML schemas, 228 schemas import database schema, database projects, 912-913 XML documents, 227 scope/access signal icons, 185 scoping options, Object Browser, 192-193 editing custom component sets, 193-194 ScriptManager, 853 ScriptManagerProxy, 853 scrollbars, extension points, 645 SDK, installing, 308-309 development environment (Azure), 1101-1104 Sealed, C#, 87 search bar, Class View window, 183-184 search folder sets, building, 216 Search In Files, 89 search results, finding, 214 searching documents, 211-212 building search folder sets, 216 Find In Files/Replace In Files, 215-216

Find Results window, 216-217 Find Symbol, 217-219 finding search results, 214 fine-tuning searches, 213-214 Incremental Search, 219-220 Ouick Find/Ouick Replace, 212-213 Replace In Files, 217 replacing text, 214-215 security, WPF browser applications, 874-876 site navigation, 876-877 Security property pages, 172 Select method, 1118 Select...Case, 104-105 selecting, data sources, 922-924 selection margin, code editor windows, 208 self-hosted services, 1002-1003 Send, 1034 SendAndReceiveReply, 1034 separating, data design from storage schema, 39-41 Sequence, 1027 server components, Server Explorer window, 189 event logs, 189 management classes, 189-190 management events, 190-191 message queues, 190 performance counters, 190 services, 191 Server Explorer, 890 SQL server databases, 890-891 adding existing databases, 890-892 Table Designer, 892 indexes, foreign keys, check constraints, 893-894 primary keys, 893 Server Explorer window, 186-187 Data Connections, 187-188

Programming, 191-192 server components, 189 event logs, 189 management classes, 189-190 management events, 190-191 message queues, 190 performance counters, 190 services, 191 server progress, showing to users, ASP.NET Ajax, 860-861 Server section, Start Options page, ASP.NET websites, 686 service references, ASP.NET Web Services, 977-979 services, 956-957 ASPNET Web Services, 959-960 ASPNET Web services and WCF. 957-959 creating and consuming, WCF (Windows Communication Foundation), 44-46 Server Explorer window, 191 SetReplySent, 1118 SetSelectionContainer, 523 SetTabPicture method, 522 Settings property page, 172 shared content, 288 finding, 289-291 installing, 291-292 storing, 291-292 types of, 288-289 SharePoint, 18 creating solutions, 24-26 sharing content, MSDN Code Gallery, 268-315 macros, 569-570 shortcut tasks, 349 signal icons, 179-181 signaling, color selection, color palette add-in, 610

Signing property page, 172 signing your work, 308-307 Silverlight 4.0. 888 Silverlight application, 18, 34, 881-882 ASP.NET websites, 686-687 creating, 882 building applications, 883-887 defining new projects, 882-884 developing for. 31-34 WPF (Windows Presentation Foundation), 33-34 simple data binding, 920-921 site housekeeping, client applications, 1063 site navigation controls, ASP.NET, 746 SiteMapPath, 746-747 site navigation, security, WPF browser applications, 876-877 SiteMapDataSource, 939 SiteMapPath, 746-747 sizing, user interfaces, ASP.NET websites, 701-702 SketchFlow, 14 skins, ASP.NET, 718-719 creating, 720-721 SliderExtender, 866 SlideShowExtender, 866 smart tags, 327 code editors. 328-329 HTML designer, 327 Office document extensions, 1095-1098 Office extension features, 1077 refactoring in code window, 396-397 Windows Forms designer, 327-328 snap lines, Windows Forms projects, 237-238 SOAP (Simple Object Access Protocol), 959 solution definition file, 148-151 Solution Explorer, 65-66, 151, 175-176 icons, 176-179 signal icons, 179-181

projects, managing, 182 solutions, managing, 181 Sort options, 185 toolbar buttons, 181 visual cues, 176-179 solution folders. 152-153 solution items, 151-153 Solution object, 512-515 Solution objects, controlling projects, 515-516 solution properties, 153-154 solutions, 145-146 building configuration properties, 156-157 creating, 146-148 solution definition file, 148-151 defined, 145 managing with Solution Explorer, 181 project dependencies, 155 solution folders, 152-153 solution items, 151-153 solution properties, 153-154 source file locations for debugging, 155-156 startup projects, 154 source file locations, for debugging, solutions, 155-156 split panes, Windows Presentation Foundation projects, 242-243 Split view, 28 SplitContainer, 786-787 splitting assemblies across multiple files, 131 SOL Azure, 1109 SQL CLR projects, starting, 916-917 SQL Editor, Query/View Designer, 903-905 SQL server databases, creating, 890-891 adding existing databases, 890-892 SQL statements, 898 creating views, 902

developing stored procedures, 902-903 debugging, 905 SQL Editor with Query/View Designer, 903-905 triggers, 906-907 user-defined functions. 907-908 writing queries, 898-900 fine-tuning, 900-901 query types, 901-902 specifying joins and join types, 900-901 SqlConnection object, 919 SolDataSource, 939 stacking behavior, controlling, 1091-1092 StackPanel control, WPF (Windows Presentation Foundation), 824-825 stage environment, deploying Azure applications to, 1139-1141 Standard controls, ASP.NET, 738-740 Standard toolbar. 62-63 standards compliance, web forms, 250 Start Action Build page, ASPNET websites, 682-683 Start Options page, Assert classes, 685 Start Debugging, 442 Start Options page ASP.NET websites Server section, 686 Start Action, 685 Assert classes, 685 Debuggers group, ASPNET websites, 686 Start Page, 52-53, 264-265 creating custom, 270-272 Get Started tab, 265-267 Guidance and Resources tab. 266-267 Latest News tab, 267 projects area, 264-266 selecting what to do at startup, 269-270 setting custom news channels, 268-269 Start View, XML Schema Designer, 229

starter kits, 288 versus templates, 292 starting threads, MSDN Forums, 279-280 Visual Studio, 52-53 startup forms, Windows Forms Application projects, 775-776 startup location, forms, 776-777 startup options, Visual Studio, 53 startup projects, solutions, 154 static constructors, 1118 static members, programming objects, 85-86 static objects, programming objects, 85-86 status bars, ToolStrip controls, 795-796 StatusStrip items, editing, 796-797 Step Into, 443-445 Step Out, 444-445 Step Over command, 441, 444 stepping through code beginning debug sessions, 441 Break All. 443 Continue, 445 ending sessions, 445-446 Run to Cursor, 441-442 Start Debugging, 442 Step Into, 443-445 Step Out. 444-445 Step Over command, 444 walking through code, 443 storage accounts, Azure, 1130-1133 storage options, Azure, 1108-1109 storage schemas, separating from data design, 39-41 stored procedures, 902-903 building and deploying, 920 C#, 917-920 debugging, 905 SQL Editor with Query/View, 903-905

stores communicating with, 1062 creating, 1061-1062 storing data in data caches, Office document extensions, 1093-1094 images, image viewer applications, 838-840 shared content, 291-292 strings, verifying with Assert classes, 377 structural matching, MEF (Managed Extensibility Framework), 643 structure of Visual Studio Wizard, 631 IDTWizard interface, 631-633 .vss and .vsdir files. 633-634 structures, creating, 91-93 style rules, 704 ASP.NET, creating, 708-709 style sheets ASP.NET, 705-707 creating, 706-708 defining attributes, CSS (Cascading Style Sheets), 232 web forms, 248-249 style toolset, ASP.NET, 706 styles ASP.NET, 703-713 applying, 711 building, 708 editing, 713 managing, 709-711 element styles, ASPNET, 704 inline styles, ASP.NET, 703-704 page-level styles, ASPNET, 704 web forms, 248-249 WPF (Windows Presentation Foundation), 824-827 styles rules, adding in CSS (Cascading Style Sheets), 232

subclassing existing controls, 801-802 subscribing to Azure services. 1127-1129 to events. 116-117 Switch, 1027 syntax, programming models, WPF (Windows Presentation Foundation), 812 syntax coloring, 321-322 System, Activities, .NET Framework, 144 System.AddIn, .NET Framework, 142 System.Collections, .NET Framework, 142 System.ComponentModel. .NET Framework. 142 System.Configuration, .NET Framework, 142 System.Data, 142 System, Diagnostics, .NET Framework, 142-143 System.Diagnostics.Contracts, .NET Framework, 143 System.Drawing, .NET Framework, 143 System.Dynamic, .NET Framework, 143 System.EnterpriseServices, .NET Framework, 143 System, Globalization, .NET Framework, 143 System.IO, .NET Framework, 143 System.Ling, .NET Framework, 143 System.Media, .NET Framework, 143 System.Messaging, .NET Framework, 143 System.Net, .NET Framework, 143 System.Security, .NET Framework, 143 System.ServiceModel, .NET Framework, 143 System.Threading, .NET Framework, 143 System, Web. , NET Framework, 144 System.Windows, .NET Framework, 144 System.Workflow.Activities, .NET Framework, 144 System.Xml, .NET Framework, 144

Τ

tab order, controls, forms, 789-790 TabContainer, 866 table definitions, modifying, 895 Table Designer, 892 indexes, foreign keys, check constraints, 893-894 primary keys, 893 Table menu, 61 table relationships building, 895-897 many-to-many, 897-898 one-to-one, 897 reflexive, 897-898 table storage, Azure, 1108 table storage data model, Azure, 1115-1120 TableLayoutPanel, 785-786 tables, web forms, 247 Table-valued function, 908 tabular data, displaying in forms, 800 tag navigation, 325 Document Outline window, 326-327 tag navigator, 325-326 tag navigator, 325-326 tags, extension points, 645-646 Target Framework, Build page, ASP.NET websites, 683 targeting your environment, projects, 54-57 task items, toggling, 528-529 Task List comment tasks, 348-350 shortcut tasks, 349 user tasks. 350 Task List window, 527-529 tasks, 527-529 Task pane, customizing, 1083-1085 Task panes, Office extension features, 1075-1076

Taskltem members, 528 TaskItem.Delete method, 527 Taskltems.Add method. 527 tasks adding, 527 removing, 527 Task List window, 527-529 Team Explorer, 12-13 Team Foundation Server. See TFS tearing down, unit tests, 373-374 templates ASPNET Web Services project templates. 960-961 ASP.NET Web Services, web service files, 961-963 for ASP.NET websites, 667-668 developing, 24 exporting, 296-297 installing, 298-299 item templates, 299-300 parameters, 294-295 versus starter kits. 292 test projects, 354-355 WCF service projects, 986-987 WCF service projects, WCF service application files, 986-987 workflow applications, 1041-1042 workflow projects, 1008-1009 Activity Designer Library, 1009 Activity Library, 1008 Console application, 1009 WCF Workflow Service Application, 1008-1009 WPF (Windows Presentation Foundation), 824-827 TerminateWorkflow, 1029 test assertions, 375 test attribute classes, unit testing framework. 370-373

test case management, TFS (Team Foundation Server), 12 test data in assertions, 381-382 test execution level, controlling setup and cleanup, 374 Test List Editor, 390-392 Test menu. 61. 357 Test Professional, 13 test project hosts, configuring, 385 test project options, unit testing, 364-365 test projects, 354, 355-356 templates, 354-355 Test menu. 357 test results managing, 365-366 viewing, 363-364 test runs, viewing, 364 test settings, controlling, 366-368 Test View window, 389-390 TestClass, 371 TestCleanup, 371 TestContext class, unit testing framework, 368-369 testing code, 353-354 unit testing. See unit testing exceptions, unit testing framework, 377-378 WCF services, 993-994 TestInitialize, 371 TestMethod, 371 TestProperty, 371 tests data-bound unit tests. See data-bound unit tests generating from existing code, 358-359 generated test code, 359-360 managing settings, 359 ordered tests, 388 organizing, 389-390

text adding to text documents, 550 editing, 550-551 replacing, 214-215 text documents, 543-545 editing, 545-550 repositioning EditPoint, 551-556 text adding, 550 editing, 550-551 text editors, 66-69, 199-200 C# code editor. 66-67 code editors. 66 debugging, 220 breakpoints, 220-222 controlling flow of running code, 222-223 editor customizations. 68-69 line wrapping, 200-202 productivity aids active hyperlinking, 320 change tracking, 319-320 coding problem indicators, 319-321 syntax coloring, 321-322 virtual space, 200-202 Visual Basic code editor, 67-68 Text windows, 524-526 TextBoxWatermarkExtender, 866 TextDocument members, 544 TextDocument methods, 544 TextPoint objects, 546 TextPoint parameters, 525 TextSelection object, 547 TFS (Team Foundation Server), 10-12, 264 build automation, 12 collaboration, 12 dashboards, 12

integration with IDEs, 12 project management, 11 reporting, 12 requirements management, 11 test case management, 12 version control, 12 web access, 12 themes, ASP.NET, 718-719 applying to sites or pages, 722-724 creating, 718-719 thread alerts, MSDN Forums, 283 Thread list, 469 threads discovering, 467-468 flagging, 467-468 inspecting, 469-471 managing, 468-469 starting on MSDN Forums, 279-280 Throw, 1029 Timeout, 371 Timer. 854 toggling task items, 528-529 Tool Window Types, 526-527 toolbar buttons, Solution Explorer, 181 toolbars Class View window, 182-183 customizing, 63-65 Debug toolbar, 439-440 navigating IDE, 62-63 Standard toolbar, 62-63 ToolStrip controls, 793-795 triggering macros, 581-582 ToolBox, 71-72, 529 Toolbox window, 527 toolboxes code snippets, 345 windows, 529-532 Windows Forms projects, 234-235

ToolBoxItems, 529 ToolBoxTabs, 529 tools code navigation tools. See code navigation tools Expression tools, 13-14 refactoring tools, 394 invoking, 395-394 Refactor menu. 395-396 smart tags, 396-397 Tools menu. 60 ToolStrip controls, 790-791 creating menus, 791-793 status bars. 795-796 toolbars, 793-795 editing StatusStrip items, 796-797 ToolStripContainer, 786-788 ToolTips, controls, forms, 790 top menu, adding to image viewer application, 833-835 tracepoints, debugging, 455-457 TransactedReceiveScope, 1034 Transaction activities, 1031-1033 TransactionScope, 1033 travel request pages, client applications, 1063-1065 TreeNode Editor, 798 triggering macros from toolbars and menus, 581-582 triggers, SQL statements, 906-907 TryCatch, 1029 Tuples, 111 type conversion, 97-99 typed DataSets, editing, 927-928 types of, shared content, 288-289

U

UDDI (Universal Description, Discovery, and Integration), 960 UI standards, form design, 773 Unit Test Wizard, 357 unit testing, 354 controlling test settings, 366-368 creating test projects, 354 templates, 354-355 generating from existing code generated test code, 359-360 managing settings, 359 generating tests from existing code, 358-359 running, tests, 362 test project options, 364-365 test projects, 355-356 Test menu, 357 test results, managing, 365-366 viewing test results, 363-364 writing, tests, 361-362 unit testing framework Assert classes, 374-376 data-bound unit tests, 378 connecting to data, 378-379 connecting with configuration files, 379-381 test data in assertions, 381-382 running and viewing data-driven results, 382-383 setup and tear down, 373-374 test attribute classes. 370-373 TestContext class, 368-369 testing exceptions, 377-378 unit tests. 353-354 ASP.NET, 383 configuring test project hosts, 385

defining attributes, 383-384 generating, 385 attributes. 386 creating ordered tests, 388 defining connections with Properties dialog box, 386-387 organizing tests, 389-390 Test List Editor, 390-392 running, 362 writing, 361-362 UnitTesting namespace, 377 Update progress control, 860 UpdatePanel, 853 UpdateProgress, 854 updating data binding, entity data model, 953-954 data with GridView. 937-938 URI (Uniform Resource Identifiers), 960 URLs, routing to controllers, 767-768 user authentication, login controls, ASP.NET, 744-745 user control events, capturing, 612-613 user controls ASP.NET, 749 adding to web pages, 750-752 creating, 749-750 color palette add-in, 606 code generation properties, 607-608 handling movement over the palette, 606-607 helper routines. 608-610 signaling color selection, 610 designing, 802-803 adding controls, 803-804 embedding, 804-805 User Interface Editor, Windows Installer, 493-494

user interfaces ASP.NET websites, 695-696 aligning items, 701-703 control positioning, 698-701 creating common look and feel, 703-724 page layout and control positioning, 696-697 sizing, 701-702 form design, 773-774 user tasks, Task List, 350 user-configurable UI, ASP.NET, 724-725 adding Web Parts to zones, 729-731 creating Web Part page, 727 defining zones, 727-729 enabling users to configure pages, 731-735 Web Part controls, 725-727 user-defined functions, 907-908

V

validation controls, ASP.NET, 740-742 ValidationSummary, 742 variables declaring, 97 inferring data type based on assignment, 118-120 viewing data in debuggers, 457 variance, 141 in generic collections, 141 VB. 85 accessing and querying data, 128 adding business logic to generated code, 126-127 adding methods to existing classes, 124-125 arithmetic operators, 101 arrays, 108-109

assignments, 100 classes, 80 collection classes, 109-111 comparison operators, 101 concatenation operators, 101 constructors, 85 creating automatically implemented properties, 133 creating custom dynamic objects, 137-138 creating events, 114-115 creating instances of nonexistent classes, 123 creating objects and setting properties with a single line of code, 120-122 creating structures, 92-93 data types, 96 declaring variables, 97 defining collections and initializing values, 122 defining constants, 99 defining interface contracts, 91 Do...While/Unit, 107 dropping underscores in line continuation, 134 dynamic objects, 139 enumerations. 86 events, handling and subscribing to, 116-117 exception handling, 113 fields and properties, 81-82 For...Each, 106 For...Next, 106 If...Then...Else, 103 inferring variables' data type based on assignment, 118-120 inheritance, 87 logical and conditional operators, 102 methods, 83 organizing code with namespaces, 93, 95

overloading members, 90 overriding behavior, 88-89 programming with attributes, 112 raising events, 116 References property page, 171-172 removing unused arguments from event handlers, 133 Select...Case. 104-105 static members and objects. 86 Tuples, 111 type conversion, 98, 99 variance. 141 writing simple, unnamed functions, 129 XML, 131-132 VBA (Visual Basic for Applications), 1073 vendor reservation services, creating, 1048-1049 verifying collections of objects. Assert classes, 377 strings, Assert classes, 377 version control, TFS (Team Foundation Server), 12 View Designer, 903-905 View menu, 58 ViewFeedback.aspx.cs, 1124-1125 viewing data in debugger, 457 Autos window, 458-459 DataTips, 460-461 Locals window, 457-458 pinning DataTips, 461-462 QuickWatch window, 459-460 visualizing data, 462-464 Watch windows, 459 watching variables, 457 data-driven results, 382-383 formal web service descriptions, 972-974 test results, 363-364

test runs. 364 web methods, 974-976 web references, 979-980 views ASP.NET MVC application, 754 functionality, 765-767 SQL statements, 902 virtual space, text editors, 200-202 VirtualPoint object, 547 Visual Basic code editor, 67-68 Visual C#, 15 Visual C++, 15 visual cues, Solution Explorer, 176-179 Visual Designers, 70 Visual F#. 15 Visual Studio installing, 47-48 choosing languages, 48 configuring development environment settings, 49-52 publishing and deploying Azure applications. 1141-1142 starting, 52-53 startup options, 53 Visual Studio 2010 Premium, 6, 9 MSDN. 10 Visual Studio 2010 Professional, 6 MSDN. 10 Visual Studio 2010 Ultimate, 6, 9-10 **MSDN**, 10 Visual Studio Agents 2010, Load Test Virtual User Packs, 13 Visual Studio automation object model, 507 Visual Studio designers, 202-203 Visual Studio documentation, 275-276 Visual Studio Express Editions, 7-8 Visual Studio Professional, 8

Visual Studio SDK, MEF (Managed Extensibility Framework), 646-653 Editor Classifier, 647-648 Editor Margin, 648-649 editor text adornment. 649-650 editor viewport adornment, 651-653 Visual Studio Tools for Office. See VSTO Visual Studio Wizard, 631 structure of, 631 IDTWizard interface, 631-633 .vss and .vsdix files. 633-634 visual styles, controls, forms, 789 visualizing, data, 462-464 vsCMElement Enumeration values, mapping, 517-519 vsCommandExecOption members, 603 VSContent file, elements of, 302-303 VSContent XML files, 301-303 vsdir files. 633-634 Add New Item Wizard, 639-640 options, 634 VSI, packaging content, 301 VSIX, 300, 301 editing manifests, 311-313 packaging content, 308-315 VSIX projects, creating, 308-310 .vss files, 633-634 Add New Item Wizard, 639-640 VSTemplates, attributes, 303-304 VSTO (Visual Studio Tools for Office), 1073-1074 smart tags, 1095

W

WAS (Windows Process Activation Services), 1003 Watch windows, 459 WCF (Windows Communication Foundation), 43 ASP.NET Web Services, 957-959 services, creating and consuming, 44-46 WCF clients, 985 WCF service application files, 986-987 WCF service applications, 985-986 WCF service projects configuring WCF services, 994-1000 consuming WCF services, 999-1002 creating WCF services, 989-993 running and testing, 993-994 hosting and deploying WCF services. 1002-1003 templates, 986-987 WCF service application files, 986-987 WCF Service, templates, 668 WCF Service Web role, 1107 WCF services, 985 configuring, 994-1000 consuming, 999-1002 creating, 989-993 debugging, 466 deploying, 1002-1003 hosting and deploying WCF services, 1002-1003 running and testing, 993-994 data-source controls, 938-939 GridView control, 936-937 selecting controls, 935-936 updating data with GridView, 937-938 WCF Workflow Service Application, 1008-1009 web access, TFS (Team Foundation Server), 12 Web controls, data binding, 935 Web Deployment Tool, 498, 499 Deploy SQL options, 500-501 launching publish proc, 502-503 Package/Publish options, 499 Web Form model, ASP.NET MVC application, 756

web forms, 244 designing, 245 browser output and validation. 249-250 controls. 245-246 editing markup, 246-247 formatting options, 247-248 standards compliance, 250 styles and style sheets, 248-249 tables, 247 web methods invoking, 976 viewing, 974-976 web pages, creating ASP.NET web pages, 687 adding control event handlers, 695 adding controls to pages, 688-691 adding page event handlers, 694 adding pages to your site, 687-688 responding to events, 690-693 Web Part controls, user-configurable UI, ASP.NET. 725-727 Web Part page, creating, 727 Web Platform Installer, 7 web references, consuming, 979-980 web role interfaces, creating (Azure), 1120-1125 web roles, Azure, 1105 configuring, 1126-1127 web service files, 961-963 web service methods, 959 web services, 959 calling, 980-983 WebMethod attribute class, 971-972 WebPartManager, 727 WebPartZone, 726, 727 WebService attribute class, 971 WebService class, 971 Web/Share deployment, 484 Web/Share execution, 484 Website menu, 59

website projects, creating, 160-161 websites adding features with AJAX. 30-31 ASP.NET. See ASP.NET websites building with ASP.NET, 26-30 enabling debugging, 423-425 WF. See Workflow Foundation (WF) WF (Windows Workflow), 43 developing application/business processes, 43-44 While, 1027 wildcards, searches, 214 Window menu, 61 Window object members, 521-522 Window panes, 524-526 Window.CommandBars property, 538 windows, 520 Autos window, 458-459 Breakpoints window, 448-449 code definition window, 224-225 code editor windows, 205-206 code pane, 206-207 indicator margin, 207-208 selection margin, 208 command window, executing commands, 532-533 Document Outline window, 326-327 Find Results window, 216-217 fonts, customizing, 77-78 interacting with, 521-524 linked windows, 536-537 Locals window, 457-458 managing, 73 docking, 74-76 IDE windows, 76-77 pinning, 73-74 output window, 533-535 Parallel Stacks window, 472

Parallel Tasks window, 475-476 QuickWatch window, 459-460 referencing, 520-521 Server Explorer window. See Server Explorer window Task List window, 527-529 Test View window, 389-390 Text windows, 524-526 Tool Window Types, 526-527 toolbox, 529-532 Watch windows, 459 Window panes, 524-526 Windows Communication Foundation. See WCF Windows Forms Application projects, 775 inheriting another form's appearance, 776 startup forms, 775-776 Windows Forms Application (WinForm), 17 Windows Forms controls, 1089 binding manually, 928 binding DataGridView, 929 binding other controls, 932 customizing cell edits, 929-931 Windows Forms designer, 327-328 Windows Forms projects adding controls to forms, 234 arranging controls, 235-236 creating, 232-233 customizing appearance, 233-234 layout grid, 236 resizing controls and editing attributes, 238 snap lines, 237-238 toolbox, 234-235 writing code, 238-241 Windows Installer, 485 Custom Actions Editor, 494-495 File System Editor, 490-491 File Types Editor, 492-493

Launch Conditions Editor, 495-498 publishing projects, 488-490 Registry Editor, 491-492 User Interface Editor, 493-494 Windows Presentation Foundation, See WPF Windows Presentation Foundation projects adding controls, 244 creating, 241-243 split panes, 242-243 Windows services, 1003 Windows Workflow, See WF WinForm (Windows Forms Application), 17 WinForms, 232, 250-251 creating new components or controls, 251-252 developing rich clients, 19-20 writing component code, 252-254 wizardResult enumeration values. 632 wizards Add New Item Wizard. See Add New Item Wizard Add-in Wizard, 586 Create Unit Tests Wizard, 359 EDM Wizard, 40-41 Explort Template Wizard, 294 finishing, 590 Import and Export Settings Wizard, 49-50 Publish Wizard, 485 Unit Test Wizard, 357 Visual Studio Wizard. See Visual Studio Wizard word wrapping, 201 worker roles, 1105-1107 workflow, 1005 components of, 1006-1008 creating, 1012 basic XAML-based workflow, 1012-1017

workflow applications creating client applications pending requests pages, 1066-1068 site housekeeping, 1063 travel request pages, 1063-1065 creating custom activity libraries, 1049 GetUsersManagerActivity custom code activity, 1051-1052 IsManager custom composite activity. 1050 creating data access libraries, 1044-1048 creating databases, 1043-1044 creating vendor reservation services. 1048-1049 designing, 1052 adding workflow persistence service. 1061 communicating with stores, 1062 creating stores, 1061-1062 data access custom activities. 1052-1054 housekeeping, 1052 workflow, 1055-1061 running, 1068-1070 scenarios, 1040-1041 templates, 1041-1042 workflow designer, 1009-1011 Workflow Foundation (WF), 1005-1006 workflow host processes, 1007 workflow persistence service, 1061 workflow projects, 1027 creating, 1012 basic XAML-based workflow, 1012-1017 dynamic workflow, 1023-1025 hosting and invoking workflows, 1017-1018 expressing workflow as an Activity Class, 1020-1023 expressing workflow as Code Activity, 1018-1020

item templates, 1011-1012 templates, 1008-1009 Activity Designer Library, 1009 Activity Library, 1008 Console application, 1009 WCF Workflow Service Application. 1008-1009 workflow designer, 1009-1011 workflow runtime engine, 1007 workflow runtime services, 1007 WorkflowApplication, 1017 WorkflowInvoker, 1017 workflows built-in workflow activities, 1026 Collection activities, 1030-1031 controlling flow, 1026-1028 error handling, 1029 managing messages, 1033-1034 Runtime and Primitives, 1028-1029 Transaction activities, 1031-1033 creating, dynamic workflow, 1023-1025 designing, 1055-1061 expressing as an Activity class, 1020-1023 expressing workflow as Code Activity. 1018-1020 Flowchart workflows, 1038-1040 hosting, 1017-1018 invoking, 1017-1018 WorkflowServiceHost, 1018 WPF (Windows Presentation Foundation), 18.34.232.807 creating browser applications, 871-873 security, 874-876 security and site navigation, 876-877 data binding, 828-830 deploying browser applications, 877-881 developing rich clients, 20-22 layout, 817-818

Canvas control, 818-819 DockPanel control, 819-821 Grid panel, 821-824 StackPanel control, 824-825 overview, 807-809 programming models, 809-811 collaboration, 812 syntax simplicity, 812 XAML (Extensible Application Markup Language), 811-812 routed events. 830-831 Silverlight and, 33-34 styles and templates, 824-827 versus XBAP 870-871 WPF applications, data binding, 932-935 WPF Browser Application, 21 WPF designer, 813-814 XAML (Extensible Application Markup Language) and design panes, 813-816 Property window, 815-816 zoom control, 815-816 WriteLine, 1029 writing code in code editors, 204-205 Windows Forms projects, 238-241 component code, WinForms, 252-254 event handlers, 577-578 macros. 571-574 macros with Macro IDE, 567-568 queries query types, 901-902 specifying joins and join types, 900-901 SQL statements, 898-900 unit tests. 361-362 WS-*, 960 WSDL (Web Service Description Language), 959 WSE (Web Service Enhancements), 960

Х

XAML (Extensible Application Markup Language), 21, 26, 823-824 programming models, WPF (Windows Presentation Foundation), 811-812 WPF designer, 813-816 Property window, 815-816 zoom control, 815-816 XAML-based workflow, creating, 1012-1017 XBAP MIME types, 879 versus WPF, 870-871 XHTML, 27 XMI .NET languages, 131-132 running against XSLT, 231 VSContent XML files, 301-303 XML documents, 226 inferring schema, 227 XML editor, 226 XML nodes, 322 XML snippet file node descriptions, 341 XML Schema Designer, 228-230 XML schemas, 227, 959 schema explorer, 228 schema views, 228 XML Schema Designer, 228-230 XML snippet format, 339-343 XmIDataSource, 939 XNA, 18 XSD (XML Schema Document), 959 XSD, VSContent file, 302 XSLT style sheets editing, 230-231 running against XML, 231

Ζ

zones

adding Web Parts to, 729-731 user-configurable UI, 727-729 zoom control, WPF designer, 815-816