

"Beyond just describing the basics, this book dives into best practices that all SharePoint developers should know."

— Arpan Shah, Director, SharePoint



SharePoint 2010 Development with Visual Studio 2010



Eric Carter
Boris Scholl
Peter Jausovec

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The .NET logo is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries and is used under license from Microsoft.

Microsoft, Windows, Visual Basic, Visual C#, and Visual C++ are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries/regions.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

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

For sales outside the United States please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data

Carter, Eric.

SharePoint 2010 development with Visual Studio 2010 / Eric Carter, Boris Scholl, Peter Jausovec.

p. cm.

Includes index.

ISBN 978-0-321-71831-0 (pbk. : alk. paper)

1. Microsoft SharePoint (Electronic resource) 2. Intranets (Computer networks) 3. Web servers. 4. Microsoft Visual studio. I. Scholl, Boris. II. Jausovec, Peter. III. Title.

TK5105.875.I6C367 2010

006.7—dc22

2010022236

Copyright © 2011 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 671-3447

ISBN-13: 978-0-321-71831-0

ISBN-10: 0-321-71831-3

Text printed in the United States on recycled paper at Edwards Brothers in Ann Arbor, Michigan.
First printing, August 2010



Contents at a Glance

Contents xi
Figures xvii
Tables xxxv
Foreword xxxix
Preface xli
Acknowledgments xliii
About the Authors xlv

1 Introduction to SharePoint 1
2 Introduction to SharePoint Development in Visual Studio 2010 105
3 Introduction to the SharePoint Object Model 153
4 SharePoint Lists 231
5 SharePoint Event Receivers 317
6 Working with Business Data 361
7 SharePoint Content Types 457
8 SharePoint Workflow 525
9 SharePoint Web Parts 579

x  **Contents at a Glance**

10 SharePoint Pages 655

11 Packaging and Deployment 699

A Preparing for SharePoint Development 787

Index 817



Contents

Figures xvii

Tables xxxv

Foreword xxxix

Preface xli

Acknowledgments xliii

About the Authors xlv

1 Introduction to SharePoint 1

What Is SharePoint Development? 1

Where Is the List Command? 2

How Do I Play My Changes Back? 5

Visual Studio to the Rescue 5

SharePoint Designer—Your SharePoint Development Sidekick 7

Introduction to SharePoint Site Collections and Sites 7

Creating a New Child Site in an Existing Site Collection 8

Content That Can Be Created in a SharePoint Site 12

Examining the Settings of a Site and Site Collection 57

The Central Administrative View of a SharePoint Server 74

A Visual Studio View of SharePoint: The Server Explorer 93

Conclusion 103

- 2 Introduction to SharePoint Development in Visual Studio 2010 105**
 - Creating a SharePoint Solution 105
 - Sandboxed Solutions versus Farm Solutions* 111
 - Exploring an Empty SharePoint Project* 113
 - Mapped Folders, Deployment, and the Hive* 117
 - SharePoint Project Items* 121
 - Exploring a SharePoint Project Item* 128
 - Features and Packages in a Visual Studio Project* 139
 - Building* 144
 - Debugging* 148
 - Conclusion 152

- 3 Introduction to the SharePoint Object Model 153**
 - Introduction 153
 - There's More Than One Object Model* 153
 - Key Objects in the SharePoint Object Model* 154
 - The Disposable Pattern in SharePoint Development 157
 - Sandboxed Solutions 160
 - A Little Set Up 162
 - From the Top: Web Applications, Site Collections, and Sites 172
 - SPContext* 172
 - SPFarm* 177
 - SPWebApplication* 179
 - SPSite* 182
 - SPWeb* 187
 - Working with Lists 194
 - SPList* 194
 - SPListItem* 201
 - SPListItemCollection* 206
 - SPFolder* 210
 - SPFile* 212
 - SPField* 217

- Working with Queries 221
 - SPQuery* 221
 - SPSiteDataQuery* 223
- Working with Users 226
 - SPUser* 226
- Other Important SharePoint Objects 227
 - SPException* 227
 - SPUtility* 228
- Conclusion 229

4 **SharePoint Lists 231**

- Introduction to SharePoint Lists 231
- Creating Lists 231
 - Building List Instances Using SharePoint Designer* 236
 - Creating List Definitions and Instances Using Visual Studio 2010* 272
 - Creating Lists Programmatically* 284
 - Programmatically Modifying Data in Lists* 300
- Conclusion 315

5 **SharePoint Event Receivers 317**

- Introduction to Event Receivers 317
 - Events in SharePoint 2010* 318
 - List Events and SPListEventReceiver* 321
 - List Item Events and SPItemEventReceiver* 322
 - List E-mail Events and SPEmailEventReceiver* 327
 - Web Events and SPWebEventReceiver* 327
 - List Workflow Events and SPWorkflowEventReceiver* 328
 - SharePoint Feature Event Receiver* 330
- Creating Event Receivers in Visual Studio 331
- Creating Feature Event Receiver Projects 341
- Deploying and Debugging Event Receiver Projects 347
- Conclusion 358

- 6 Working with Business Data 361**
 - Introduction to Business Connectivity Services 361
 - Creating External Content Types Using SharePoint Designer 362
 - Creating External Content Types Using Visual Studio 384
 - Build a Simple External Content Type in Visual Studio* 386
 - Build a CRUD-enabled External Content Type* 401
 - Store the Connection Information and Make It Accessible During Runtime* 421
 - Build Associations Between External Content Types* 429
 - Custom Properties* 450
 - Deployment Considerations for External Content Types Using Visual Studio* 452
 - Conclusion 455

- 7 SharePoint Content Types 457**
 - Introduction to Content Types 457
 - Content Type Definition* 458
 - SharePoint Site Columns* 459
 - Content Type Hierarchy* 464
 - Content Types in Visual Studio 2010 470
 - Creating a List Definition from a Content Type* 487
 - Advanced Content Types 491
 - Word Documents and Content Types* 491
 - Content Types and Workflows* 521
 - Conclusion 522

- 8 SharePoint Workflow 525**
 - Workflow Basics 525
 - Workflow Activities 528
 - Control Flow Activities* 529
 - Container Activities* 529
 - Standard Activities* 529
 - SharePoint Workflow Activities* 532

SharePoint Workflows	536
<i>How Does Workflow Deployment Work?</i>	547
<i>List, Site, and Content Type Associations</i>	548
Workflow Forms	552
Workflows in SharePoint Designer	562
<i>Importing SharePoint Designer Workflow Into Visual Studio 2010</i>	570
Conclusion	578

9 SharePoint Web Parts 579

Introduction to Web Parts	579
Web Part Fundamentals	580
Creating a Visual Web Part	581
<i>Creating Web Parts with AJAX That Update without Having to Reload the Entire Web Page</i>	597
<i>Web Part Customization</i>	600
Creating a Code-only Web Part (Sandboxed)	609
Creating a Silverlight Web Part	619
<i>Create a New Silverlight Application</i>	621
<i>Deploying a Silverlight Web Part to SharePoint</i>	630
Web Part Connections	636
Configuration of Web Parts	643
<i>Connection Strings</i>	643
<i>Code Access Security (CAS)</i>	647
Conclusion	654

10 SharePoint Pages 655

Introduction to SharePoint Pages	655
SharePoint Architecture	655
Customizing and Developing SharePoint Pages	661
<i>Customizing and Creating Master Pages</i>	661
<i>Creating Site Pages</i>	672
<i>Creating Application Pages</i>	682
Conclusion	697

11	Packaging and Deployment	699
	The Basics and Beyond	699
	SharePoint Features	699
	<i>SharePoint Elements</i>	706
	<i>Deploying SharePoint Features</i>	709
	SharePoint Solution Packaging	712
	<i>Deploying SharePoint Solutions</i>	717
	The SharePoint Project Structure in Visual Studio	721
	<i>Feature Designer</i>	725
	<i>Mapped Folders</i>	733
	<i>Feature Activation Dependencies</i>	739
	<i>Feature Designer for Advanced Users</i>	743
	<i>Package Designer</i>	745
	<i>Packaging Explorer</i>	747
	<i>Package Designer for Advanced Users</i>	748
	Configurable Deployment	751
	<i>Custom Deployment Steps</i>	760
	<i>Deployment Conflicts</i>	774
	<i>Sandboxed and Farm Solutions</i>	777
	<i>Auto-Retract</i>	779
	<i>Silverlight Debugging</i>	779
	Conclusion	786
A	Preparing for SharePoint Development	787
	Supported Operating Systems	787
	Installing SharePoint 2010	788
	<i>Pre-Installation on a Client Operating System</i>	789
	<i>Pre-Installation on a Server Operating System</i>	794
	<i>Installation and Configuration of SharePoint 2010</i>	796
	Installing Visual Studio 2010	803
	Installing SharePoint Designer 2010	807
	Creating a Site Collection	809
	Developer Dashboard	812
	Conclusion	815
	<i>Index</i>	817



Foreword

Over the last decade, we've seen SharePoint evolve from a collaboration application to a business collaboration platform. With SharePoint's growing popularity, organizations have looked to extend SharePoint beyond traditional collaboration to scenarios such as human workflow, document processing, line-of-business portals, Internet sites, and more. Before the arrival of SharePoint 2010, developers spent a great deal of effort developing SharePoint applications and going through a number of manual steps and processes.

With the SharePoint 2010 release, we really focused on the developer audience by investing in out-of-the-box tools and features to make the SharePoint development experience familiar. We received very clear feedback from developers that they wanted an experience similar to developing using ASP.NET applications, including out-of-the-box Visual Studio templates and an F5 experience for rapid development, debugging, and deployment. We listened and delivered with the SharePoint 2010 and Visual Studio 2010 releases.

I fundamentally believe that with the release of SharePoint 2010 the number of SharePoint developers will double. That's a significant jump given that we estimate hundreds of thousands of developers have worked on a SharePoint project in the last six to twelve months. This growth will be driven by the expected increase in SharePoint customers along with the business opportunity this will bring to developers everywhere; it's a great time to be a SharePoint developer.





Whether you're a .NET developer with no SharePoint experience or an experienced SharePoint 2003 or 2007 developer, this book will help you take advantage of the rich SharePoint 2010 platform and Visual Studio 2010 tools! Beyond just describing the basics, this book dives into some of the best practices that all SharePoint developers should know. This book is special because it's written by experts Boris, Eric, and Peter—people with deep real world experience *and* product insight: a SharePoint developer book by SharePoint developers!

—Arpan Shah
Director, SharePoint
June 2010



Preface

In late 2007, we had just put the final touches on Visual Studio 2008. For the Visual Studio Tools for Office (VSTO) team, the 2008 release was an important one. Visual Studio 2008 was the team's first major investment in SharePoint projects. Visual Studio 2008 included two projects to build SharePoint Workflow: the Sequential and State Machine Workflow Projects.

In some ways, however, our Visual Studio 2008 investment in SharePoint development was too little and too late. All around us, SharePoint development was taking off. When we attended Microsoft developer conferences and talked about VSTO, the level of interest in SharePoint was unbelievable. Although we had a great set of developer tools for the Office client in VSTO, SharePoint support in Visual Studio was still very limited.

During this time, a group of us began to seriously survey the SharePoint development space to determine how we could catch up. This space is vast and a large part of our effort was to figure out what would be possible to do in the next two years to best support SharePoint developers. In addition, a vibrant ecosystem for SharePoint development was springing up all around us, so we knew that it was also important to have a great extensibility model to allow third parties to plug into Visual Studio. Our focus was on delivering a solid set of basic support for SharePoint development, particularly in making it easy to build and deploy SharePoint solutions.

This book tries to put in one place all the information you need to succeed using Visual Studio to program against SharePoint Foundation. The first three chapters form an introduction to SharePoint development, including

extensive coverage of the SharePoint UI and how to extend SharePoint, an introduction to Visual Studio, and an introduction to the SharePoint object models. The remaining chapters take you through Visual Studio's support for creating the various elements of a SharePoint solution, including lists, workflow, and connections to business data.

This book is unique in that it focuses on Visual Studio throughout the book where most books only mention Visual Studio in a few chapters. Our focus is on showing you what Visual Studio provides to make SharePoint development easy. We also introduce SharePoint Designer as a tool to fill in some of the gaps left by Visual Studio, showing you ways to easily create some SharePoint elements.

The goal of this book is to allow you to be productive in SharePoint development by relying on Visual Studio—it won't require you to have a deep understanding of everything going on behind the scenes in SharePoint because Visual Studio takes care of a lot of this for you. Visual Studio really makes SharePoint development much more accessible to a new developer, similar to how MFC and Visual Basic made Windows development accessible to a broader range of developers.

This book provides an insider view of all the rich features of Visual Studio that were created to enable you to develop SharePoint solutions. We participated in the design and implementation of many of these features. Therefore, we can speak from the unique perspective of living and breathing Visual Studio support for SharePoint for the past two years. Programming SharePoint with Visual Studio is powerful and fun. We hope you enjoy using Visual Studio to develop SharePoint solutions as much as we enjoyed writing about it and creating it.

—Eric Carter, Boris Scholl, and Peter Jausovec

■ ■ 2 ■ ■

Introduction to SharePoint Development in Visual Studio 2010

Visual Studio 2010 provides the templates and tools you need to develop complete SharePoint solutions. In this chapter you will create your first SharePoint solution and we will introduce you to some of the projects, project item templates, and tools that are in Visual Studio 2010 for SharePoint development.

Creating a SharePoint Solution

First, make sure you have followed the setup instructions in Appendix A, “Preparing for SharePoint Development.” Once your machine is set up, you will need to launch Visual Studio as an administrator. The SharePoint projects in Visual Studio require administrator privileges to interact with SharePoint. To launch Visual Studio as an administrator, locate the Microsoft Visual Studio 2010 shortcut in the Start Menu under All Programs > Microsoft Visual Studio 2010. Right click on the Microsoft Visual Studio 2010 shortcut. You can choose Run as administrator from the context menu to run Visual Studio as an administrator.

Alternatively, if you just want to make Visual Studio start up with administrator privileges every time you launch it, you can change the Microsoft Visual Studio 2010 shortcut properties to always run as administrator. To do this, right click on the Microsoft Visual Studio 2010 shortcut and choose Properties. Click the Compatibility tab as shown in Figure 2-1. Then check the Run this program as an administrator check box and press OK.

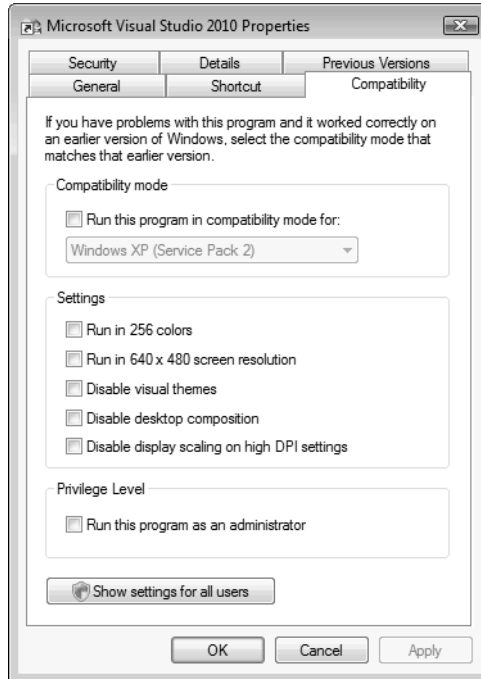


Figure 2-1: Setting the Microsoft Visual Studio 2010 shortcut to start as administrator

Now use the modified Microsoft Visual Studio 2010 shortcut or the Run as administrator command in the context menu to launch Visual Studio with administrator privileges. Once Visual Studio has started up, choose New > Project... from the File menu. This brings up the New Project dialog. Select Visual C# as the language from the tree view control on the left. Then expand the Visual C# node and select SharePoint under the Visual C# node. Expand the SharePoint node and click the 2010 node. This will display all the available SharePoint project types as shown in shown in Figure 2-2.

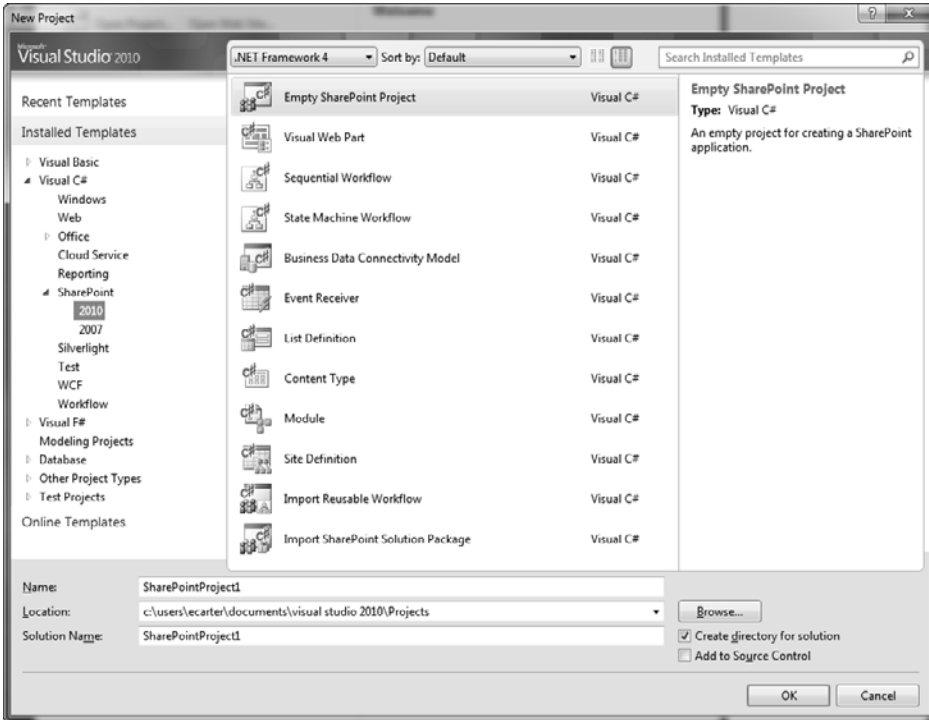


Figure 2-2: SharePoint projects in the New Project dialog in Visual Studio

Visual Studio has 12 basic SharePoint project types; they are listed in Table 2-1. The most basic SharePoint project type is the empty SharePoint Project. This project begins empty but lets you create and add any SharePoint project items you want to it as you go along. So in an empty SharePoint project you could add a web part, a list definition, and so on.

TABLE 2-1: SharePoint Project Types

Project Type	Description
Empty SharePoint Project	An empty SharePoint project; as with all projects in this table, you can add and remove arbitrary SharePoint item types once they are created.
Visual Web Part	A SharePoint project prepopulated with a visual web part project item; web parts are controls written using ASP.NET that users can place in a SharePoint web part page.

Continues

TABLE 2-1: SharePoint Project Types (Continued)

Project Type	Description
Sequential Workflow	A SharePoint project prepopulated with a Sequential Workflow project item; when you create a workflow, you are writing a program that is structured as a multistep process that may be long running, waiting for humans or external systems to complete tasks or other processes before continuing. A sequential workflow proceeds in a linear top-to-bottom fashion.
State Machine Workflow	A SharePoint project prepopulated with a State Machine Workflow project item; a state machine workflow uses concepts such as states, events, and transitions to model more complex business processes with multiple possible paths through the workflow.
Business Data Connectivity Model	A SharePoint project prepopulated with a Business Data Connectivity Model project item; a business data connectivity model allows you to define a data model and code that lets you integrate external data into SharePoint.
Event Receiver	A SharePoint project prepopulated with an Event Receiver project item; an event receiver handles events that are raised by lists and other key objects in SharePoint. For example, an event receiver could execute custom code when a new item is added to a SharePoint list.
List Definition	A SharePoint project prepopulated with a List Definition project item; a list definition defines the schema of a list along with other things that define the list, such as views and forms used by the list.
Content Type	A SharePoint project prepopulated with a Content Type project item; a content type defines the schema for an item in a list—examples of content types in SharePoint include the Contact content type and the Announcement content type. Content types can also be associated with document libraries, allowing you to create a specialized document type, such as an expense report content type complete with a document template, such as an Excel workbook or Word document, to be used when a new expense report is created.
Module	A SharePoint project prepopulated with a Module project item; a module is used when creating deployments to provision the SharePoint site with one or more files that are needed by a solution—for example, you might use this to deploy a custom document to a document library or some other custom resource like an image file or an application page.

TABLE 2-1: SharePoint Project Types (*Continued*)

Project Type	Description
Site Definition	A SharePoint project prepopulated with a Site Definition project item; a site definition is a template that is used when creating a new SharePoint site and defines all the initial content in that site—for example, the subsites that we saw in Chapter 1, Table 1-7 are all backed by custom site definitions.
Import Reusable Workflow	A SharePoint project created by importing a workflow that was initially created in SharePoint Designer.
Import SharePoint Solution Package	A SharePoint project created by importing a .WSP SharePoint Solution Package exported from SharePoint or SharePoint Designer.

A second class of SharePoint project is prepopulated with one particular SharePoint project item type. This class includes the Visual Web Part, Sequential Workflow, State Machine Workflow, Business Data Connectivity Model, Event Receiver, List Definition, Content Type, Module, and Site Definition projects. These projects are empty SharePoint projects that have one SharePoint project item preadded to them—the item type specified by the project type designation. So, for example, the Content Type SharePoint project is an Empty SharePoint project with a Content Type project item preadded to it. As with the Empty SharePoint project type, you can continue to add SharePoint project item types to this class of projects or even remove from them the initially preadded SharePoint project item type.

A third class of SharePoint projects is populated by a wizard that runs when the project is first created. This class of SharePoint projects includes the Import Reusable Workflow and Import SharePoint Solution Package. The Import Reusable Workflow project creates a SharePoint project by importing a workflow from a workflow created in SharePoint or SharePoint Designer. The Import SharePoint Solution Package project creates a SharePoint project by importing a .WSP file exported from SharePoint Designer.

It is worth noting at this point that you have two general options for how you structure your SharePoint solutions. You can use a single project

and add as many SharePoint project items as you need to that project. Or you can divide your solution into multiple projects. A key limitation of a project is that a single project can only produce a single .NET assembly (a .DLL file). So in cases where you need to factor your solution to produce multiple .NET assemblies you need to create multiple projects. You might also choose to divide your solution into multiple projects if you are developing a solution with other developers. This can make it easier for a solution to be simultaneously worked on by multiple developers.

Later in this chapter we will discuss in more detail what each of these project items and projects do. For now, let's start by creating an Empty SharePoint Project. As shown in Figure 2-2, you can specify a name and location for your project. You can also optionally add the project to source control by checking the Add to Source Control check box. Once you've set the name and location for your project, click the OK button to create the project.

A second dialog will appear to configure the site URL and security level for the newly created project as shown in Figure 2-3. The site URL should designate the SharePoint site where you want to deploy and test your SharePoint solution. As discussed in Chapter 1, a particular machine can host multiple SharePoint sites. If you click the Validate button, Visual Studio will verify that the URL you have typed corresponds to an already created SharePoint site.

In order for you to successfully deploy and debug your SharePoint solution, you must also have the proper permissions on the SharePoint site you designated with the site URL. Your user account must be added as the Site Owner or Site Collection Administrator for the site URL. In some cases, a customization you build may be deployed at the site level and at other times a customization may only be deployable at the site collection level. For more information on setting up the SharePoint site properly so you can deploy and debug, see Appendix A, "Preparing for SharePoint Development."

In the same dialog shown in Figure 2-3, you must choose the trust level for the SharePoint solution. After the project is created, you can change the trust level later by using the Properties window for the project as shown in Figure 2-5 (see page 114). The property to change is the Sandboxed Solution property—if this is set to False the solution will have a Farm solution trust level.

Sandboxed solutions run in a secure, monitored process. Sandboxed solutions can be deployed without requiring SharePoint administrative privileges. If you choose a sandboxed solution, you can only use project item types that are valid in sandboxed solutions.

If you choose the option “Deploy as a farm solution,” Visual Studio will deploy the solution as a fully trusted farm solution. If you choose a farm solution, you can use all available SharePoint project item types in your project, but deployment will require administrative privileges and the solution will run in full trust.

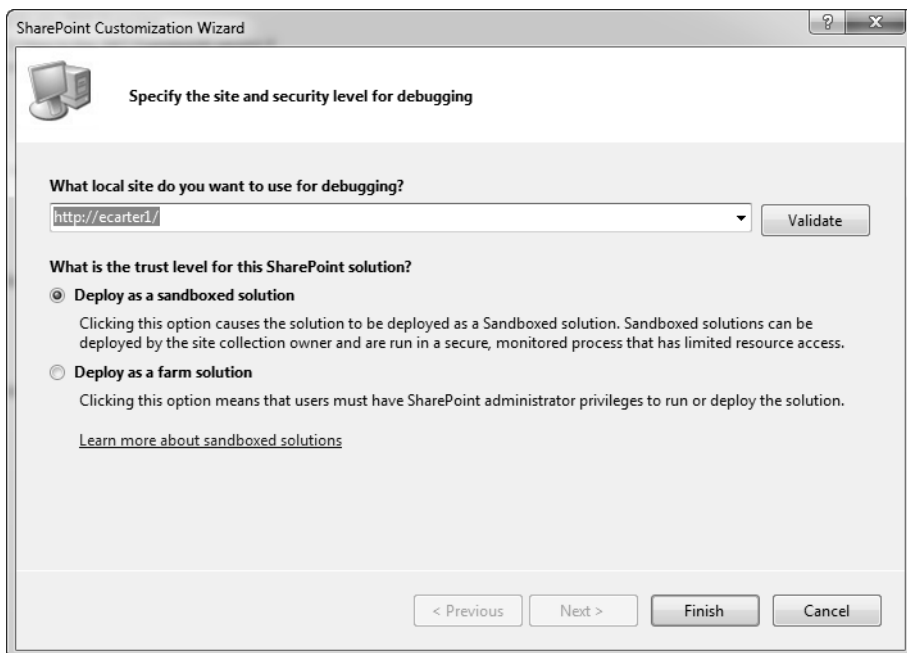


Figure 2-3: Debugging options for empty projects

For this example, choose Deploy as a sandboxed solution. Click the Finish button as shown in Figure 2-3 to complete the creation of the project.

Sandboxed Solutions versus Farm Solutions

Early in the project creation process, Visual Studio asks you to decide between using a sandboxed solution or a farm solution. It is worth considering in more

detail the difference between a sandboxed solution and a farm solution and when to choose one over the other.

Prior to SharePoint 2010, all solutions you could create were farm solutions. In Chapter 1 we saw that SharePoint solutions are deployed to a farm that could consist of one to many servers. Each server in the farm can have multiple web applications running on it. A web application can in turn have one or more site collections, and a site collection has one or more sites. Farm solutions can impact the entire SharePoint system and are available to all site collections and sites in the farm. This is sometimes desirable, but sometimes can have undesired effects because a farm solution that is misbehaving can impact all sites and site collections in the system.

In SharePoint 2010, you can create a new type of solution called a sandboxed solution. Sandboxed solutions are deployed at the site collection level rather than the farm level, so this lets you isolate a solution so it is only available to one site collection within the farm. Sandboxed solutions also run in a separate process from the main SharePoint IIS web application process, and the separate process is throttled and monitored with quotas to protect the SharePoint site from becoming unresponsive due to a misbehaving sandboxed solution.

It is worth mentioning that sandboxed solutions solve an organizational problem as well—in many organizations it is difficult to get permission to install a farm solution because of the possible impact that could have on the SharePoint system. System administrators in charge of running a SharePoint site have been reluctant in the past to allow custom solutions to run on their sites. With the advent of SharePoint 2010, there is now a robust system in place to monitor and throttle these custom solutions so that system administrators don't have to worry about a custom solution bringing the entire SharePoint site down. In addition, with sandboxed solutions, users can upload solutions without requiring administrator approval.

So if sandboxed solutions are so great, why are farm solutions still around at all in SharePoint 2010? Well, because of the need to restrict and throttle a sandboxed solution so that it cannot negatively impact the entire site, there are restrictions on the kinds of solutions you can build with a sandboxed solution. The most significant restrictions disallow cre-

ation of application pages, visual web parts, or code-based workflows with a sandboxed solution. You can, however, create a web part without using the visual designer and deploy it in a sandboxed solution—we will see how to work around this particular limitation in Chapter 9, “SharePoint Web Parts.”

So in the end, the choice between sandboxed and farm solutions should come down to whether or not you need to create an application page or a workflow with code in it. For these kinds of solutions, you should pick a farm solution. For all other solutions, pick a sandboxed solution. The only other reason to use a farm solution over a sandboxed solution is if you really have some code that needs to run at the web application or farm level, perhaps because it needs to interact with or move data between multiple site collections. In this case, you would create a farm solution as well.

Exploring an Empty SharePoint Project

Returning now to the project we just created, let’s inspect the structure of an empty SharePoint project as shown in Figure 2-4. First, click on the root node, Solution ‘SharePointProject1’ in our example. In the Properties win-

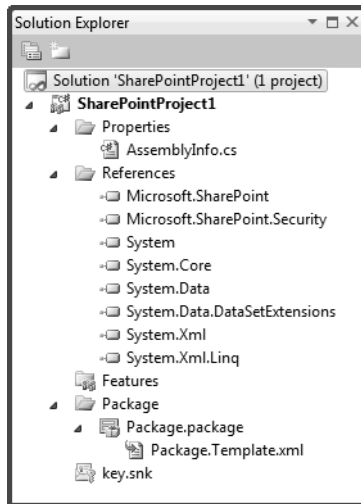


Figure 2-4: An empty SharePoint project

dow, you will see the properties associated with the solution. The two most interesting properties are the Active config and the Startup project properties. Active config sets whether to build debug or release assemblies. By default this starts out as Debug | Any CPU. Typically, during development you will use debug, but when you are ready to deploy the solution you will use Release | Any CPU setting. The Startup project will set which project's startup settings will be used when you press F5 when there are multiple projects in the solution. Since in typical solutions all projects will be deploying to the same SharePoint site URL, this won't matter much in practice unless you are building one solution that creates multiple deployments.

Now, click on the Project node, SharePointProject1, in our example. In the Properties window are a number of SharePoint specific properties as shown in Figure 2-5. Table 2-2 describes the functions of each of these properties. Two of the properties you configured during project creation can be changed here: Site URL—which designates the SharePoint site where you deploy and test your SharePoint solution, and Sandboxed Solution—which when set to True indicates that the solution will be a sandboxed solution and when set to False indicates that the solution will be a farm solution.

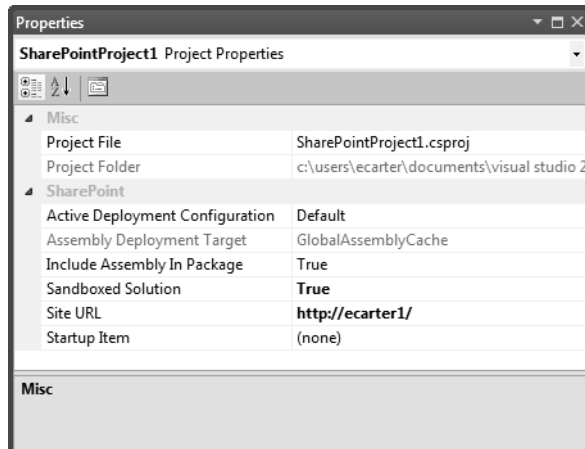


Figure 2-5: SharePoint project Properties window

TABLE 2-2: SharePoint Project Properties

Project Properties	Description
Project File	The name of the Visual Studio project file.
Project Folder	The name of the folder where the project file is saved.
Active Deployment Configuration	If set to Default, this does the standard deployment when deploying a SharePoint solution. If set to No Activation then when a solution is deployed to the SharePoint site, Visual Studio won't try to activate the solution. For more information, see Chapter 11, "Packaging and Deployment."
Assembly Deployment Target	Sets where the assembly created by this project will be deployed; for sandboxed solutions, this property is disabled because in a sandboxed solution the management of where the assembly is loaded and runs is controlled by SharePoint. For a farm solution, this can be GlobalAssemblyCache or WebApplication. For some SharePoint solutions, the assembly must go in the global assembly cache, for others it can be put at the web application level in a directory specific to the web application. For more information see Chapter 9, "SharePoint Web Parts."
Include Assembly in Package	Sets whether the assembly created by this project is packaged into the .WSP file created when the project is built; you will typically set this to True unless you have a project that is only used for deploying CAML-based SharePoint artifacts like a list definition.
Sandboxed Solution	If set to True, the project will deploy as a partially trusted sandboxed solution, if set to False, the project will deploy as a fully trusted farm solution.
Site URL	The URL for the SharePoint site where you want to deploy and test your SharePoint solution.
Startup Item	The item in the project that will be started when you press F5. For example, if you have an application page in your project, it will open the application page in the browser when you press F5.

Next, consider the Properties folder in the Solution Explorer. In this folder, you will find an `AssemblyInfo.cs` file that contains the attributes that will be added to the assembly that is created when the project is built. Almost all of these attributes are identical to the ones you would find when creating a simple class library project. The only one that is new is the `AllowPartiallyTrustedCallers` attribute. This attribute is used for partially trusted solutions (sandboxed solutions or farm solutions that have Assembly Deployment Target set to `WebApplication` as we saw in Table 2-2). For projects that have Assembly Deployment Target set to `GlobalAssemblyCache`, the `AllowPartiallyTrustedCallers` attribute can be removed.

The References folder in the Solution Explorer contains all the referenced assemblies for the project. The set of referenced assemblies are identical to the ones you would find when creating a simple class library project with two additions: `Microsoft.SharePoint` and `Microsoft.SharePoint.Security`. The `Microsoft.SharePoint` assembly contains the server object model for SharePoint that you will use when writing code against SharePoint. The `Microsoft.SharePoint.Security` assembly contains code access security objects that are used for partially trusted solutions.

The Features folder is a special folder that is found only in a Visual Studio SharePoint project. This folder contains SharePoint features that have been created for the project. A SharePoint project can contain zero or more features. By default, when you add a new SharePoint item to a SharePoint project that requires a feature, Visual Studio will create a new feature automatically or reuse an existing feature if there is already a feature in the project with the same scope (Farm, Site, Web, or Web Application). For more information on working with features, see Chapter 11, “Packaging and Deployment.”

The Packages folder is another special folder that is found in SharePoint projects. This folder contains information that allows the project to create a package file or `.WSP` file. Package files are used to deploy SharePoint features to a SharePoint site. By default, when you add a new SharePoint item to a SharePoint project that results in the creation of a new feature, that new feature will automatically be added to the package file associated with the project. For more information on working with packages, see Chapter 11, “Packaging and Deployment.”

The next file you will find in a SharePoint project is the `key.snk` file. This is a strong name key that is used to sign the output assembly of the project.

Mapped Folders, Deployment, and the Hive

One item you will often find in a SharePoint project that isn't found in our example is a mapped folder. Mapped folders give you a way to take resources and other files in your project and add them to folders in the Visual Studio project that are mapped to file system locations where those files need to be deployed on the SharePoint server. For example, imagine you have an application page you have developed that needs to deploy a file to the SharePoint server's images folder. To do this you would right click on the Project node and choose Add, then SharePoint Images Folder. This creates a mapped folder in the project called Images. Any folders you add to the images folder will be created on disk (if they aren't already there) and the contents of those folders will be copied to the SharePoint server's images folder when the project is deployed.

It is time for another aside regarding SharePoint terminology. We've just implied that SharePoint has an images folder—what is this and what other special folders does SharePoint have? When you build a deployment for SharePoint you build a SharePoint package, which is basically a CAB file (like a ZIP file if you aren't familiar with the CAB format) that has in it a set of files and instructions that are used to install your SharePoint solution. The instructions are encapsulated in one or more SharePoint Feature files, which consist of XML markup that is read at install time. A special program called `stsadm.exe` takes the SharePoint package file (which is a CAB file with a `.WSP` extension) and reads the SharePoint feature files in the package to determine how to install the SharePoint solution. These SharePoint feature files in turn can refer to additional files that are packaged within the SharePoint package. `Stsadm.exe` then does two major things—it adds information to the SharePoint content database and it copies files to the file system. So a SharePoint solution typically modifies the SharePoint content and configuration databases and adds files to the file system of the SharePoint server machine.

There are three general locations where SharePoint copies files to the file system of the server during deployment. The first location is the global assembly cache of the server machine. Solutions that have assemblies that need full trust will copy to this directory when Assembly Deployment Target is set to GlobalAssemblyCache as we saw in Table 2-2.

The second location is directories specific to a web application (which is an IIS concept we described in Chapter 1). One of those web application-specific directories is the bin directory. This is where assemblies are deployed if you set the Assembly Deployment Target property to WebApplication. To determine where the web application directory is, launch the Internet Information Services (IIS) manager on the SharePoint server (use the search box in the Start menu to search for it). Once you've launched the IIS manager, expand the Sites folder and find the web application you are interested in—in a default install it will be called SharePoint -80 as shown in Figure 2-6. Right click on the SharePoint -80 node, and pick Explore from the context menu. This will open the base directory for your web application as shown in Figure 2-7. Of interest here are several directories and files you may use. The web.config file is used to configure ASP.NET specific settings—you have to modify this file for some SharePoint development scenarios we will see later in this book. The bin folder is the bin directory associated with the web application where assemblies are sometimes deployed. There are other directories here that are used for web part development, such as the wpre-resources folder.

The third location of interest for deployment is known in the SharePoint developer world as *the hive*, which is the location on disk where SharePoint installs feature definitions, site definitions, and other content used to provision the web site. SharePoint builds on its own extensibility model—many of the features in the SharePoint web site correspond to actual files you can inspect and learn from in these directories. The hive can be found at Program Files\Common Files\Microsoft Shared\Web Server Extensions\14. Some of the folders found in the hive are shown in Figure 2-8.

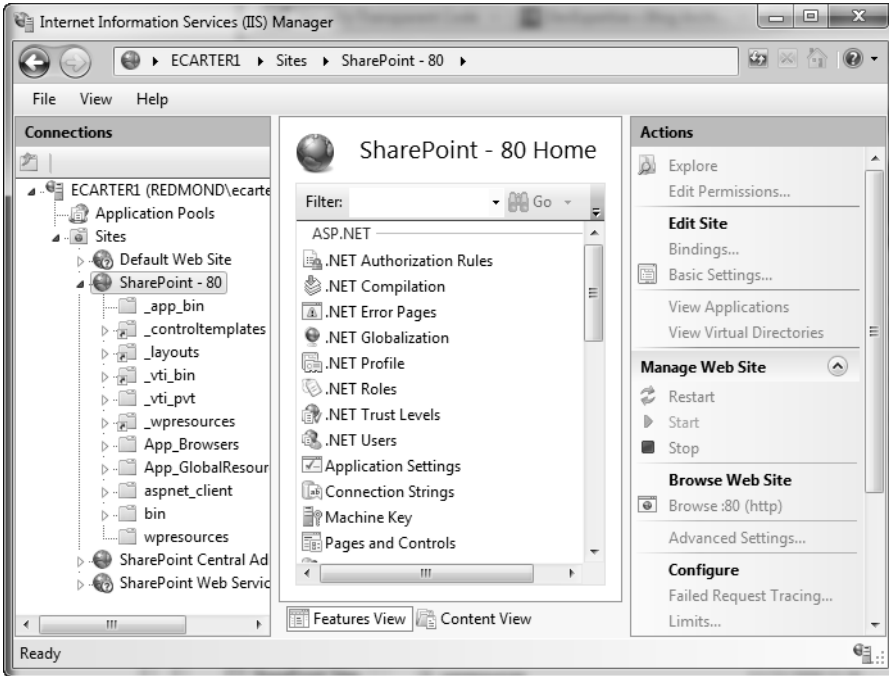


Figure 2-6: The Internet Information Services (IIS) Manager showing the web application SharePoint -80

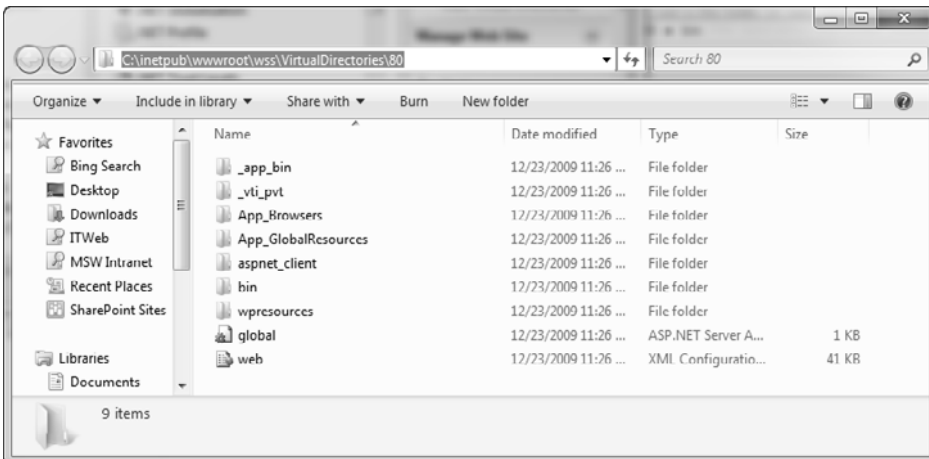


Figure 2-7: Directories and files associated with the SharePoint -80 web application

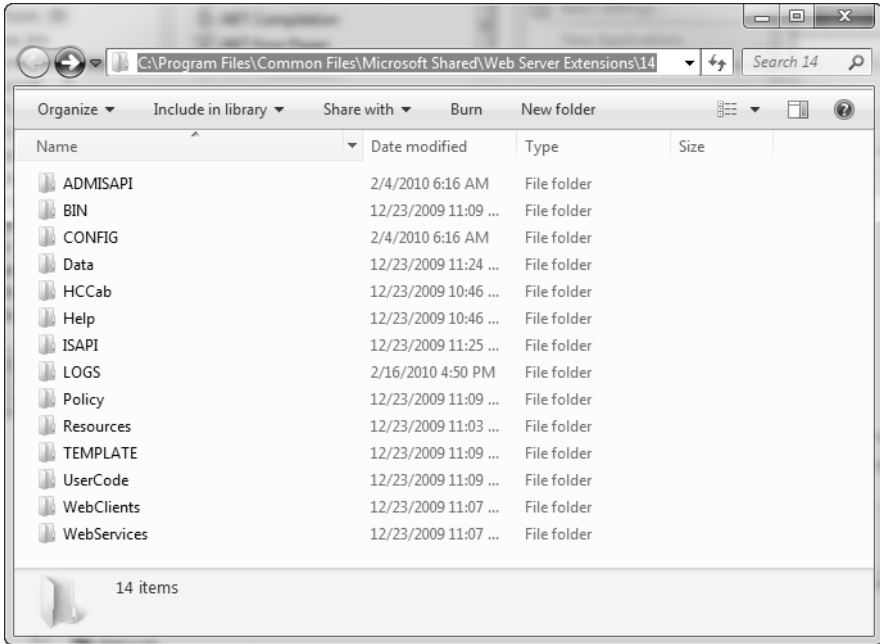


Figure 2-8: Directories and files in the SharePoint hive

When you add a mapped folder in Visual Studio by right clicking on the Project node and choosing Add, then SharePoint Mapped Folder, you will see the dialog shown in Figure 2-9, which lets you view all the folders in the hive to which you might want to deploy items. In Figure 2-9, we have expanded the TEMPLATE folder, which is the main place to which you will deploy items. In this folder, you can see there is an IMAGES folder, where you can deploy arbitrary images you want to use from web parts or application pages. There are other directories as well—for example, the SiteTemplates folder, where you install Site Definitions files and the LAYOUTS folder, where you can find the master page being used for the SharePoint server. You will typically create a subdirectory within the LAYOUTS folder if you want to install your own application pages. We will learn more about the hive throughout this book.

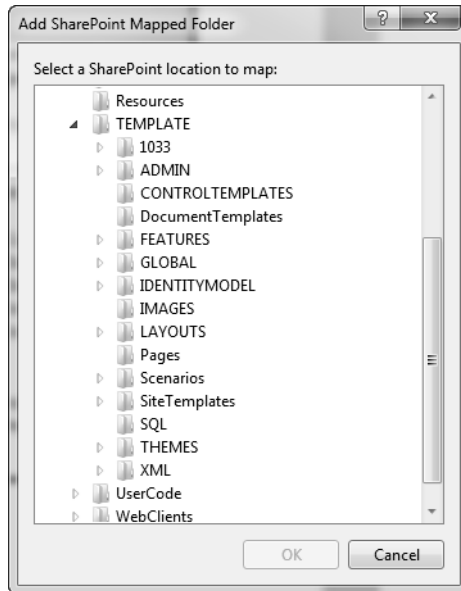


Figure 2-9: Adding a Mapped Folder in Visual Studio to the Layouts folder

SharePoint Project Items

So now that we've seen the basic structure of an empty SharePoint project and learned a little bit more about deployment, let's consider what happens when we add a SharePoint project item to the SharePoint project. To add a SharePoint project item, right click on the Project node in Server Explorer (titled SharePointProject1 in our example) and choose Add, then New Item... from the context menu. The Add New Item dialog shown in Figure 2-10 appears. There are a number of SharePoint project items that can be added to a SharePoint project. Table 2-3 lists the project item types and briefly describes each one. It also lists the chapter in this book where each project item type is described in detail.

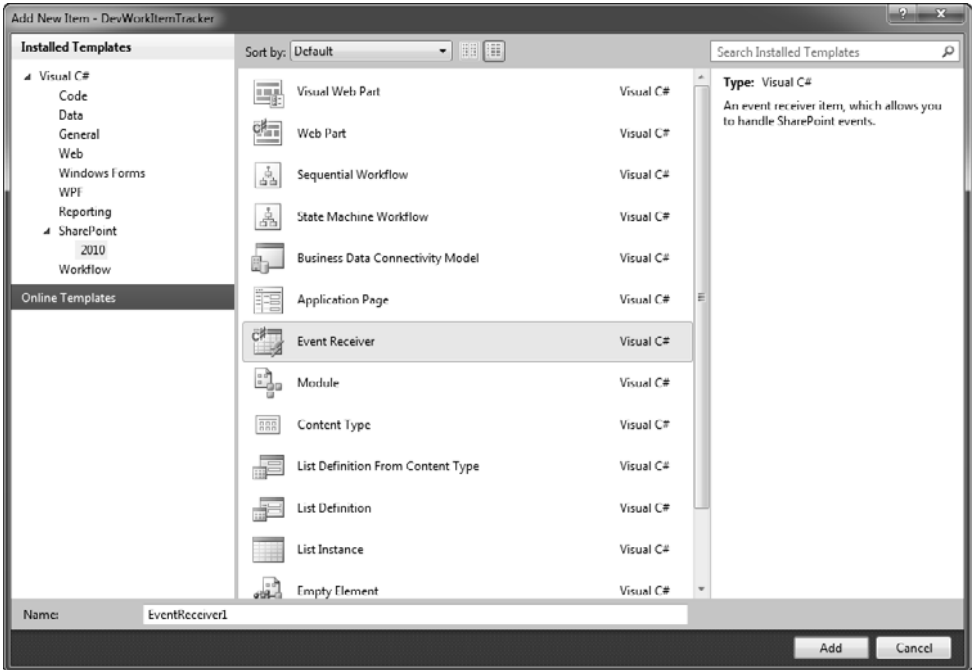


Figure 2-10: Add New Item dialog

TABLE 2-3: SharePoint Project Item Types

Project Item Type	Description	Described in Chapter
Application Page	An ASP.NET web page that is displayed within a SharePoint site	Chapter 10, "SharePoint Pages"
Business Data Connectivity Model	A business data connectivity model allows you to define a data model and code that lets you integrate external data into SharePoint.	Chapter 6, "Working with Business Data"
Content Type	A content type defines the schema for an item in a list—examples of content types in SharePoint include the Contact content type and the Announcement content type. Content types can also be associated with document libraries, allowing you to create a specialized document type such as an "Expense	Chapter 7, "SharePoint Content Types"

TABLE 2-3: SharePoint Project Item Types (Continued)

Project Item Type	Description	Described in Chapter
Content Type <i>(continued)</i>	Report" content type complete with a document template (e.g., an Excel workbook or Word document) to be used when a new expense report is created.	Chapter 7, "SharePoint Content Types"
Empty Element	An empty element is used to create a SharePoint project item that has a single XML file called Elements.xml associated with it. In this file you can define SharePoint elements that aren't natively supported yet by Visual Studio, such as a site column for a list. This element can then be referenced by other project items and installed with the rest of the solution.	Chapter 11, "Packaging and Deployment"
Event Receiver	An event receiver handles events that are raised by lists and other key objects in SharePoint. For example, an event receiver could execute custom code when a new item is added to a SharePoint list.	Chapter 5, "SharePoint Event Receivers"
List Definition	A list definition defines the schema of a list along with other things that define the list, such as views and forms used by the list.	Chapter 4, "SharePoint Lists"
List Definition from Content Type	Allows you to create a list definition based on a content type already in the project or on the local SharePoint server	Chapter 4, "SharePoint Lists"
List Instance	A list instance allows you to create an instance of a list as part of your solution; for example, you could have a project with a custom list definition called Expense Reports and two instances of that list, one called "International Expense Reports" and one called "Domestic Expense Reports."	Chapter 4, "SharePoint Lists"
Module	A module is used when creating deployments to provision the SharePoint site with one or more files that are needed	Chapter 11, "Packaging and Deployment"

Continues

TABLE 2-3: SharePoint Project Item Types (Continued)

Project Item Type	Description	Described in Chapter
Module (continued)	by a solution—for example, you might use this to deploy a custom document to a document library or some other custom resource (e.g., an image file or an application page). A module has an Elements.xml file that you can edit to specify the file or files associated with the module.	Chapter 11, “Packaging and Deployment”
Sequential Workflow	A multistep process that can be long running, waiting for humans or external systems to complete tasks or other processes before continuing. A sequential workflow proceeds in a linear top-to-bottom fashion.	Chapter 8, “SharePoint Workflow”
State Machine Workflow	A state machine workflow uses concepts such as states, events, and transitions to model more complex business processes with multiple possible paths through the workflow.	Chapter 8, “SharePoint Workflow”
User Control	User controls are ASP.NET controls that can be deployed and reused by application pages or web parts. For example, you might create a custom user control that provides a drop-down control with custom behavior you want to reuse in several application pages and web parts.	Chapter 9, “SharePoint Web Parts”
Visual Web Part	Web parts are controls written using ASP.NET that users can place in a SharePoint web part page. The Visual Web Part project gives you a user control that you can visually edit combined with a class deriving from ASP.NET’s web part class that hosts the user control.	Chapter 9, “SharePoint Web Parts”
Web Part	Web parts are controls written using ASP.NET that users can place in a SharePoint web part page. The Web Part project does not use ASP.NET user controls or the visual designer for an ASP.NET user control; instead it has you edit a class deriving from ASP.NET’s web part class.	Chapter 9, “SharePoint Web Parts”

Table 2-3 shows the SharePoint project items available when you right click on the Project node item and choose Add New Item. In addition, there are four more SharePoint project items that are only available when you right click on an existing item in a SharePoint project and then choose Add > New Item... . It can be confusing that these items don't appear if you have the Project node selected but only when you have an existing project item selected. But this reinforces the idea that these project items only make sense when used in conjunction with other SharePoint project items. These four additional project items are listed in Table 2-4.

TABLE 2-4: SharePoint Project Item Types Dependent on Other Project Item Types

Project Item Type	Description	Described in Chapter
Business Data Connectivity Resource Item	Dependent on Business Data Connectivity (BDC) Model; allows you to add a resource file to your BDC solution to provide localized names for a BDC model.	Chapter 7, "SharePoint Content Types"
Global Resources File	Dependent on having other SharePoint project items in the project; allows you to add a resource file to provide localized names that can be used by any feature or element file in the solution.	Chapter 11, "Packaging and Deployment"
Workflow Association Form	Dependent on having a Sequential or State Machine workflow in the project; an ASP.NET form that can be displayed when a workflow is associated with a SharePoint list, document library, or site. This can be used to configure initial settings for the workflow.	Chapter 9, "SharePoint Web Parts"
Workflow Initiation Form	Dependent on having a Sequential or State Machine workflow in the project; an ASP.NET form that is displayed when an instance of a workflow is begun for an item in a list or document library or for a site workflow.	Chapter 8, "SharePoint Workflow"

For the purposes of this chapter, let's create an event receiver. An event receiver handles events that are raised by lists, items in a list, webs (SharePoint sites), and workflow. To create an event receiver, select Event Receiver from the list of project item types that can be added, as shown in the Add New Item dialog in Figure 2-10. Accept the default name (EventReceiver1) and press the Add button. The SharePoint Customization Wizard dialog appears. Here you can choose the type of event receiver you want to create, as shown in Figure 2-11.

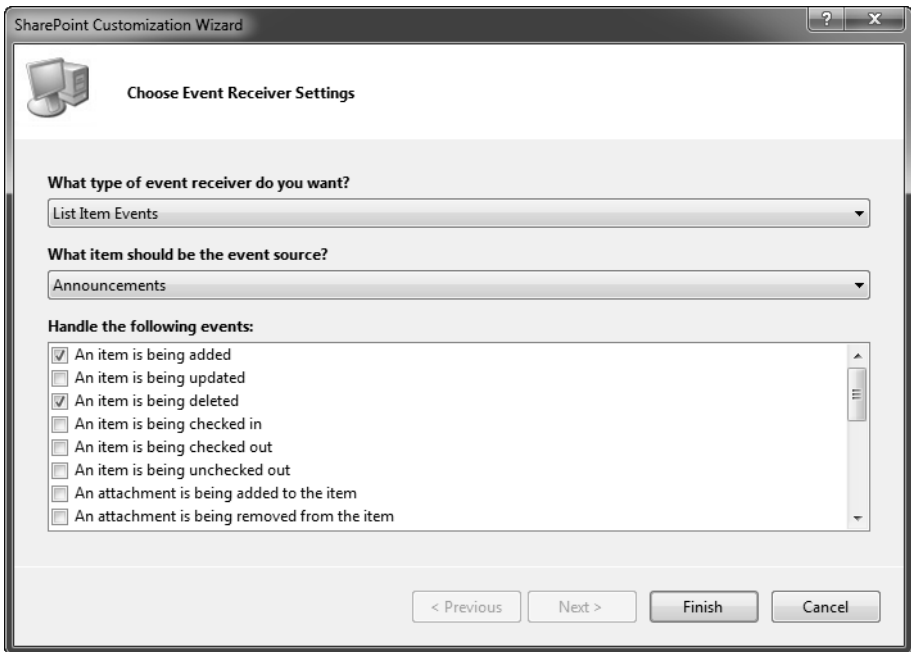


Figure 2-11: The SharePoint Customization Wizard dialog configured to handle list item events for the Announcements list

There are five types of event receivers you can create: a list event receiver, a list item event receiver, a list e-mail event receiver, a web (SharePoint site) event receiver, and a list workflow event receiver. List item, list workflow, and list e-mail event receivers act on a specific list instance that you must also elect in the dialog in Figure 2-11. List and web event receivers pertain to a web scope and act on the current SharePoint site to which you are deploying.

For this example, let's create an event receiver that handles list item events for the Calendar list associated with the SharePoint site. To do this, pick List Item Events from the first drop-down in the dialog and select Announcements as the specific list for which we will handle events. After making these selections, the list box at the bottom of the dialog shows the different list item events that can be handled by your code. Check the check boxes next to An item is being added, An item is being deleted, An item was added, and An item was deleted. The dialog should look like Figure 2-11. Then press the Finish button.

Visual Studio adds the SharePoint project item representing an event receiver. The resulting project structure now looks like Figure 2-12.

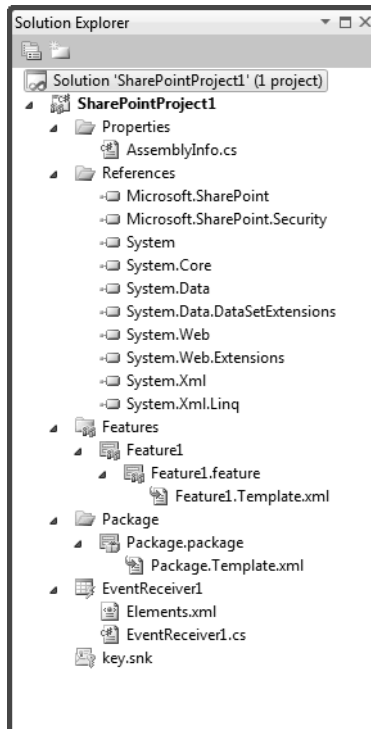


Figure 2-12: A SharePoint project with a single SharePoint project item for an Event Receiver

Exploring a SharePoint Project Item

Now that we have added our first SharePoint project item to this project, we will explore some of the new things we can see in the project with the addition of a SharePoint project item. First note that there is a project item icon in Solution Explorer titled `EventReceiver1`, and parented under that icon are two additional items: an XML file called `Elements.xml` and a code file called `EventReceiver1.cs`. As we will see throughout this book, this is a common pattern for many SharePoint project items.

The root SharePoint project item node “`EventReceiver1`” has properties that configure the event receiver. This root node is actually a folder that contains two files. The `Elements.xml` file contains the XML that is used to describe the customization to SharePoint and is read by `stsadm.exe` as part of installing and configuring the solution. The `EventReceiver1.cs` file contains custom code that defines the behavior of the SharePoint customization and will be compiled into an assembly to be deployed to the SharePoint site. Let’s consider each of these files in more detail.

The Elements.xml File

The first file to consider is the `Elements.xml` file. This is sometimes referred to as an Element Manifest file, and is an XML file that contains information that describes the SharePoint item being created to SharePoint, in this case an event receiver. Behind the scenes, Visual Studio will refer to this `Elements.xml` file in a feature file it has created. The feature file in turn is contained by a package—a package can contain one or more features as shown in Figure 2-13. When Visual Studio deploys the package, each feature file and associated `Elements.xml` file will be copied to the SharePoint server. SharePoint will read the feature file that will refer to the `Elements.xml` file. The `Elements.xml` file, as we will see, in turn refers to event handlers defined in an assembly. Once SharePoint has read the feature file and associated `Elements.xml` and assembly files, it can make the feature available for activation in the SharePoint site. We will consider the Visual Studio project support for features and packages in more detail later in this chapter and in Chapter 11, “Packaging and Deployment.” Note that in this diagram, one feature has custom code associated with it represented by a

.NET assembly. It is possible for multiple features to use code written within the same assembly.

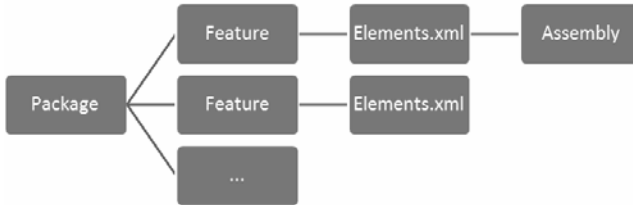


Figure 2-13: The relationship among a package, a feature, element manifests, and an assembly

When you click on the Elements.xml file node in the Solution Explorer, you will see several properties in the Properties window that can be configured as shown in Figure 2-14. Note that the properties in the Properties window change if you click in the contents of the Elements.xml file, so be sure you’ve clicked on the node in the Solution Explorer tree view. These properties are organized into three categories: Advanced, Misc, and SharePoint. Let’s consider the properties in each of these three categories.

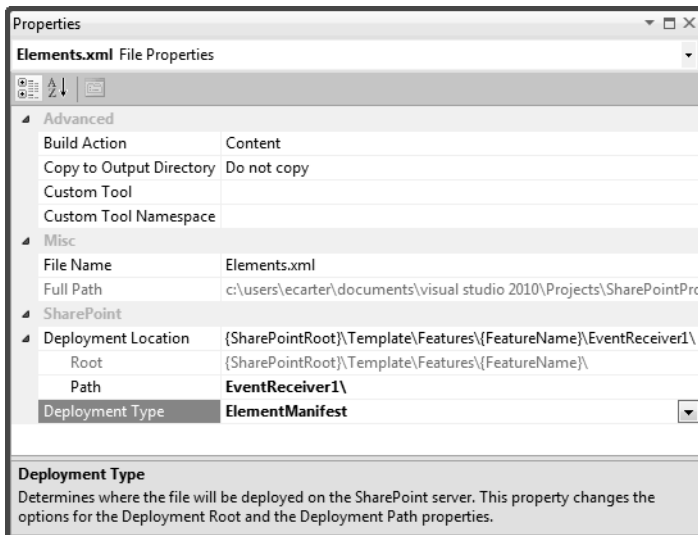


Figure 2-14: The properties associated with the Elements.xml file node

Under the Advanced category of properties there are four properties: Build Action, Copy to Output Directory, Custom Tool, and Custom Tool Namespace. These properties tell Visual Studio how to process the Elements.xml file and you should just leave these properties set to their original values—Build Action is set to Content, Copy to Output Directory is set to Do not Copy, and the other two properties have no setting.

Under the Misc category of properties there are two properties—the File Name and the Full Path to the file so you can locate where it is stored on disk. As with the Advanced properties, there is no good reason to change these properties; they should be left set to their original values.

Finally, the SharePoint category of properties includes the property Deployment Location with child properties Root and Path and the property Deployment Type. The Deployment Location property tells you where on the SharePoint server the Elements.xml file will be deployed when you build and deploy the solution. In our example, it is set to {SharePointRoot}\Template\Features\{FeatureName}\EventReceiver1\. In our example, {SharePointRoot} is a substitution token that will be replaced by the actual root file location where SharePoint features are deployed on a server, typically a path such as "C:\Program files\Common Files\Microsoft Shared\web server extensions\14\", although SharePoint could be installed to a different drive or program file location than in this example. Another term you will hear used for this set of directories found under {SharePointRoot} is *the SharePoint hive*. {FeatureName} is another substitution token that will be replaced by the name of the feature that this SharePoint project item file is associated with, in our example: Feature1.

The Deployment Type property is set to ElementManifest—this reflects that Elements.xml is an element manifest and must be deployed to a folder corresponding to the feature with which the SharePoint project item file is associated. Changing this property would be a bad idea for this file because it would change the location where the file is deployed to one not appropriate for an element manifest.

Now that we've considered the properties associated with the Elements.xml file node, double click on the Elements.xml file node in the Solution Explorer window to open the contents of the Elements.xml file. The contents of the file are shown in Listing 2-1. Note that this is representative of

the contents of an Elements.xml file for an event receiver. But for other SharePoint project item types—for example a list definition—the contents of the Elements.xml file will look quite different. This can be confusing to new SharePoint developers because they think that all Elements.xml files have similar contents when in truth, the contents of Elements.xml are specific to the SharePoint project item type.

Listing 2-1: The Elements.xml file defining event receivers for four list item events

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
<Receivers ListTemplateId="104">
  <Receiver>
    <Name>EventReceiver1ItemAdding</Name>
    <Type>ItemAdding</Type>
    <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
    <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
    <SequenceNumber>10000</SequenceNumber>
  </Receiver>
  <Receiver>
    <Name>EventReceiver1ItemDeleting</Name>
    <Type>ItemDeleting</Type>
    <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
    <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
    <SequenceNumber>10000</SequenceNumber>
  </Receiver>
  <Receiver>
    <Name>EventReceiver1ItemAdded</Name>
    <Type>ItemAdded</Type>
    <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
    <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
    <SequenceNumber>10000</SequenceNumber>
  </Receiver>
  <Receiver>
    <Name>EventReceiver1ItemDeleted</Name>
    <Type>ItemDeleted</Type>
    <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
    <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
    <SequenceNumber>10000</SequenceNumber>
  </Receiver>
</Receivers>
</Elements>
```

The contents of the Elements.xml file have a root element called Elements. Elements has a child element called Receivers. Receivers contains a

Receiver element for each of the four event receivers we defined for our Event Receiver project item. Each Receiver element contains 5 subelements listed in Table 2-5. You might also notice the ListTemplateId attribute, which is set to 104. To find out where this magic number comes from, use the Server Explorer that we saw in Chapter 1 to browse the site. Under Team Site, List Templates, Communications select the Announcements list template. In the Properties window, you will see that its Type_Client property is 104. The number 104 tells SharePoint to associate the event receiver with the Announcements list definition.

TABLE 2-5: Subelements Contained within the Receiver Element in Elements.xml

Element	Description
Name	The name of the receiver
Type	The type of the event handler; this name is defined by SharePoint and must also match the name of the method defined in the class handling the event.
Assembly	The full name of the assembly where the event handler for this event is defined; Visual Studio allows a token to be used here, <code>\$SharePoint.Project.AssemblyFullName\$</code> , which will be replaced with the full name of the assembly when the project is built.
Class	The fully qualified name of the class where the event handler for this event is defined, e.g., <code>SharePointProject1.EventReceiver1.EventReceiver1</code>
SequenceNumber	The number inside the SequenceNumber element defines the order in which events are executed. If there are two or more event receivers and they are handling the same events, the one that has the smallest number in the SequenceNumber element is executed first.

As you might imagine, the Elements.xml or element manifest for other SharePoint project item types contains different content that defines that particular SharePoint project item type. Every Elements.xml file, regardless of type, has a root element called Elements, however.

The Code File (EventReceiver1.cs)

Below the SharePoint project item node “EventReceiver1” you will see a code file called EventReceiver1.cs, shown in Listing 2-2. This contains a class that derives from Microsoft.SharePoint.SPIItemEventReceiver. Event handlers are added with calls to the base class implementation of the event handler. Note that the names of these event handlers map to the names used in the Elements.xml file.

Listing 2-2: EventReceiver1.cs

```
using System;
using System.Security.Permissions;
using Microsoft.SharePoint;
using Microsoft.SharePoint.Security;
using Microsoft.SharePoint.Utilities;
using Microsoft.SharePoint.Workflow;

namespace SharePointProject1.EventReceiver1
{
    /// <summary>
    /// List Item Events
    /// </summary>
    public class EventReceiver1 : SPIItemEventReceiver
    {
        /// <summary>
        /// An item is being added.
        /// </summary>
        public override void ItemAdding(SPIItemEventProperties properties)
        {
            base.ItemAdding(properties);
        }

        /// <summary>
        /// An item is being deleted.
        /// </summary>
        public override void ItemDeleting(SPIItemEventProperties properties)
        {
            base.ItemDeleting(properties);
        }

        /// <summary>
        /// An item was added.
        /// </summary>
        public override void ItemAdded(SPIItemEventProperties properties)
        {

```

```
        base.ItemAdded(properties);
    }

    /// <summary>
    /// An item was deleted.
    /// </summary>
    public override void ItemDeleted(SPItemEventProperties properties)
    {
        base.ItemDeleted(properties);
    }
}
}
```

We will add some code to the `ItemDeleting` and `ItemAdded` event as shown in Listing 2-3. In `ItemDeleting`, we first call the base `ItemDeleting` method. Then we use the `properties` parameter that is passed to the function and use the `ListItem` property. The `ListItem` property is a parameterized property to which we can pass a field name to read and write a field from the item being deleted from the list. We use the `ListItem` property to access the `Title` field and append the `Title` field with an asterisk. Next, we call the `Update` method on the `ListItem` to update the item being deleted with the new `Title`. Finally, we use the `Cancel` property on the `properties` parameter to cancel the deletion of the item. We can stop an item from being deleted in the `ItemDeleting` event because this event is fired before the item is deleted—the `ItemDeleted` event happens after the item is deleted, so putting this code into that event would not work.

We also add code to the `ItemAdded` event. Here, we again use the `ListItem` property on the `properties` parameter passed to the event. We modify the `Title` to add the string “ added” to it after it is added to the list. We then call the `Update` method on the `ListItem` to update the item that was added so our change will be shown in the list.

Listing 2-3: EventReceiver1.cs with custom event handlers for `ItemDeleting` and `ItemAdded`

```
using System;
using System.Security.Permissions;
using Microsoft.SharePoint;
using Microsoft.SharePoint.Security;
using Microsoft.SharePoint.Utilities;
using Microsoft.SharePoint.Workflow;
```

```
namespace SharePointProject1.EventReceiver1
{
    /// <summary>
    /// List Item Events
    /// </summary>
    public class EventReceiver1 : SPItemEventReceiver
    {
        /// <summary>
        /// An item is being added.
        /// </summary>
        public override void ItemAdding(SPItemEventProperties properties)
        {
            base.ItemAdding(properties);
        }

        /// <summary>
        /// An item is being deleted.
        /// </summary>
        public override void ItemDeleting(SPItemEventProperties properties)
        {
            base.ItemDeleting(properties);
            SPWeb web = properties.OpenWeb();
            properties.ListItem["Title"] =
                properties.ListItem["Title"] + "*";
            properties.ListItem.Update();
            properties.Cancel = true;
        }

        /// <summary>
        /// An item was added.
        /// </summary>
        public override void ItemAdded(SPItemEventProperties properties)
        {
            base.ItemAdded(properties);
            SPWeb web = properties.OpenWeb();
            properties.ListItem["Title"] =
                properties.ListItem["Title"] + " added";
            properties.ListItem.Update();
        }

        /// <summary>
        /// An item was deleted.
        /// </summary>
        public override void ItemDeleted(SPItemEventProperties properties)
        {
            base.ItemDeleted(properties);
        }
    }
}
```

The Root Project Item Node or Folder (EventReceiver1)

When you click on the root level SharePoint project item node EventReceiver1, you will see several properties in the Properties window that can be configured as shown in Figure 2-15. First note that the Properties window indicates that EventReceiver1 is a folder—the Properties window says Folder Properties. The properties for EventReceiver1 are organized into three categories: Misc, SharePoint, and SharePoint Events. Let's consider the properties in each of these three categories.

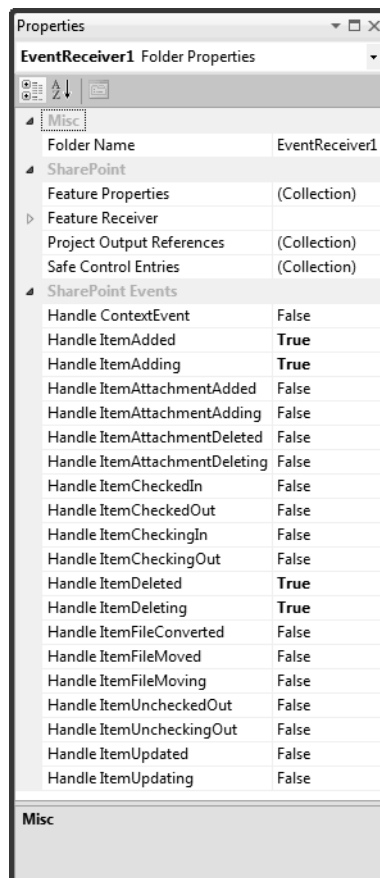


Figure 2-15: The properties associated with the EventReceiver1 node

Misc Properties The sole property under the Misc category in Figure 2-15 is Folder Name. If you change the name of the project item node—either by changing the Folder Name property or by renaming the project item node in the Solution Explorer, it will automatically change some but not all areas in the project that refer to the project item node. For example, if you were to rename EventReceiver1 to EventReceiver2, Visual Studio automatically fixes up the feature “Feature1” associated with the SharePoint project item to refer to the new project item name. But it doesn’t change the names of the files contained in the (now EventReceiver2) folder or any of the classes that were created. So after changing the node to EventReceiver2, the code file is still titled EventReceiver1.cs and the class inside the code file is still titled EventReceiver1. More critically, the Elements.xml file still refers to EventReceiver1.

You could manually rename EventReceiver1.cs to EventReceiver2.cs and even refactor the class contained in the newly renamed EventReceiver2.cs file to be EventReceiver2. In this case, the Elements.xml file will be updated correctly. But in some cases the Elements.xml file will not be updated correctly after a refactor. For example, if you change the namespace that your class is defined in from SharePointProject1.EventReceiver1 to SharePointProject1.MyEventReceiver, the Elements.xml file will not be updated to have the right fully qualified class names in the Class elements. You would have to manually update the Elements.xml file to ensure it contains the new SharePointProject1.MyEventReceiver namespace or the project won’t run successfully.

SharePoint Properties Continuing with our exploration of the properties associated with the root level SharePoint project item node EventReceiver1, you will see a number of properties under the category SharePoint in Figure 2-15. Feature Properties is a collection of key value pairs that are used when deploying the feature associated with the event receiver to SharePoint. These properties are deployed with the feature and can be accessed later in SharePoint using the SPFeaturePropertyCollection object. So for example, you might use feature properties to associate some configuration information or other static data with your feature.

The Feature Receiver set of properties includes the subproperties Assembly and Class Name. You can use these properties to specify an assembly and class name that you want to handle events that are raised when the Feature associated with your EventReceiver1 item is raised. You can create a class in your current solution and refer to it or include another assembly in your SharePoint package—for more information on how to include additional assemblies in your SharePoint package, see Chapter 11, “Packaging and Deployment.”

Events that your feature receiver can handle include FeatureInstalled, FeatureUninstalling, FeatureActivated, and FeatureDeactivating. So you could write code that runs when your EventReceiver1 is installed and uninstalled from the SharePoint site, maybe to add additional resources or lists required on install and remove those additional resources or lists on uninstall. We will discuss feature event receivers more in Chapter 5, “SharePoint Event Receivers.”

Project Output References are used to tell Visual Studio about any dependent assemblies your project item requires to run. For example, maybe your event receiver uses a helper class library called HelperLibrary.dll. You can use the Project Output References project to tell Visual Studio about the helper class library and then Visual Studio will package the dependent assembly in the final solution. For more on Project Output References, see Chapter 11, “Packaging and Deployment.”

Finally Safe Control Entries is used to designate whether an ASPX control or web part is trusted and can be used by users on the site. In the context of an Event Receiver, this property is not applicable and doesn’t do anything. We will see this property used in Chapter 9, “SharePoint Web Parts.”

SharePoint Events Properties In Figure 2-15, under the category SharePoint Events are a number of properties that correspond to the list of events we saw earlier in the wizard shown in Figure 2-11. For the events that we checked earlier, the properties Handle ItemAdded, Handle ItemAdding, Handle ItemDeleted, and Handle ItemDeleting are set to True. To add and remove events that this event receiver handles you can change which SharePoint event “Handle” properties are set to True or False. To see the

impact of setting a property that was previously set to False to True, open the Elements.xml file and the EventReceiver1.cs files under the EventReceiver1 node and arrange your windows so these files are visible while you change “Handle” properties under the SharePoint Events category to True or False. You will notice that setting a property like Handle ItemCheckedIn, which was previously False to True, adds a block of XML to the Elements.xml file and adds a new event handler to the EventReceiver1.cs file. If you then set that property back to False, you will see that Visual Studio removes the block of XML that it previously added from the Elements.xml file, but leaves the event handler it added to the EventReceiver1.cs file intact. It leaves the code in EventReceiver1.cs intact because you might have written some code in the handler and Visual Studio wants to preserve any code you wrote. Also, having an inactive event handler in EventReceiver1.cs (inactive because it isn’t registered in the Elements.xml file) will have no ill-effect on your remaining active event handlers (active because they *are* registered in the Elements.xml file).

Features and Packages in a Visual Studio Project

We’ve now explored the properties and files that are associated with a new SharePoint project item. We’ve seen the Elements.xml file, the code file associated with an event receiver, and the properties associated with each of these files and the root EventReceiver1 folder for the SharePoint project item.

You may have noticed that when we added the event receiver project item to our blank solution, some new items appeared under the Features folder. Let’s examine the Features and Package folders in the SharePoint project to start to get an idea of what Visual Studio does to package and deploy our SharePoint solution.

Features

Just to make things a little more interesting, let’s create a second event receiver. Follow the steps we did earlier in the chapter to create a second event receiver called EventReceiver2. For the second event receiver, choose List Item Events as the type of event receiver to create, use Calendar as the event source, and handle the event when an item is being added.

Now double click on the project item called Feature1 under the Features folder. The Visual Studio Feature designer appears as shown in Figure 2-16. Note that we now have two event receivers in our solution, EventReceiver1 and EventReceiver2, and Feature1 is configured to install both event receivers.

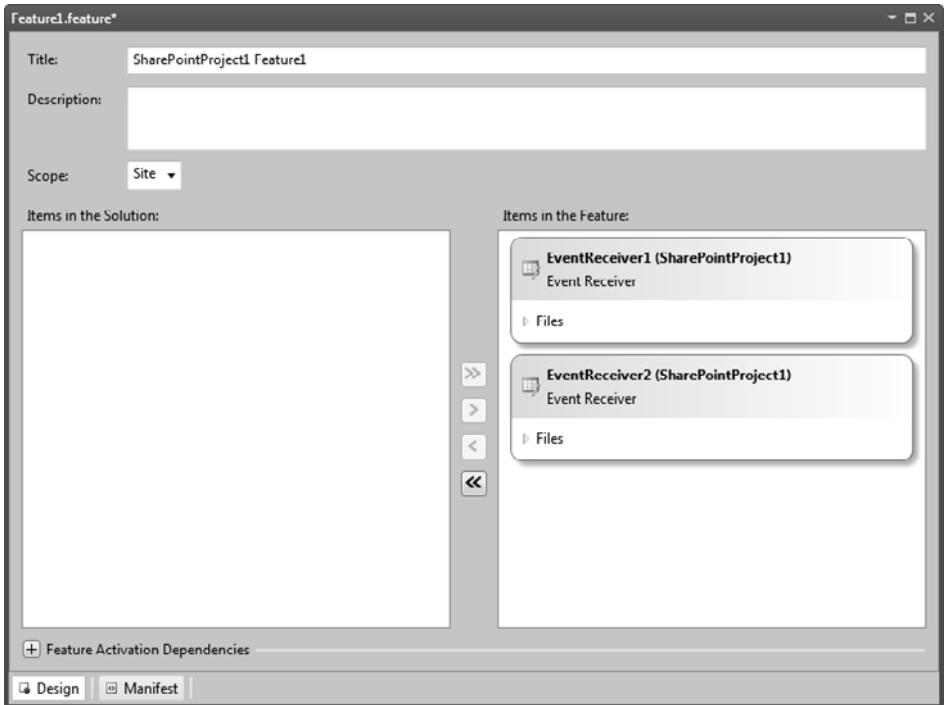


Figure 2-16: Visual Studio’s Feature designer with two SharePoint project items to deploy

It is possible to add features to the Features folder. For example, maybe you want EventReceiver1 and EventReceiver2 to be deployed as separate features. You could create a separate feature called “Feature2” and install EventReceiver1 in Feature1 and EventReceiver2 in Feature2. Doing this would enable the event receivers to be installed and uninstalled separately. Another reason you might need to have separate features is when you have SharePoint project items you want to deploy that need to be installed at a different scope. If you drop down the Scope drop-down in Figure 2-16, you can see that a feature can be installed to one of four scopes: Farm, Site (the Site Collection Level), Web (the Site level), and Web-

Application (all sites hosted by an IIS web application). Due to historical reasons, SharePoint sometimes uses the word Site to refer to a Site Collection and Web to refer to a SharePoint Site.

Let's create a second feature by right clicking on the Features folder and choosing Add Feature. A new Feature called Feature2 is created. In the Feature designer that will appear for Feature2, click on the EventReceiver2 SharePoint item and click the > button to move the feature from the left-hand list to the right-hand list. Then back in the Feature1 designer, ensure that EventReceiver2 is not installed by Feature1 by clicking on EventReceiver2 and pressing the < button to move it from the right-hand list to the left-hand list. The resulting Feature1 designer is shown in Figure 2-17. This shows that Feature1 will now install only EventReceiver1 not EventReceiver2. The right-hand list contains the features that will be installed; the left-hand list contains other items in the solution that have not been added to this feature. Also in Figure 2-17, we have expanded the Files outline and the

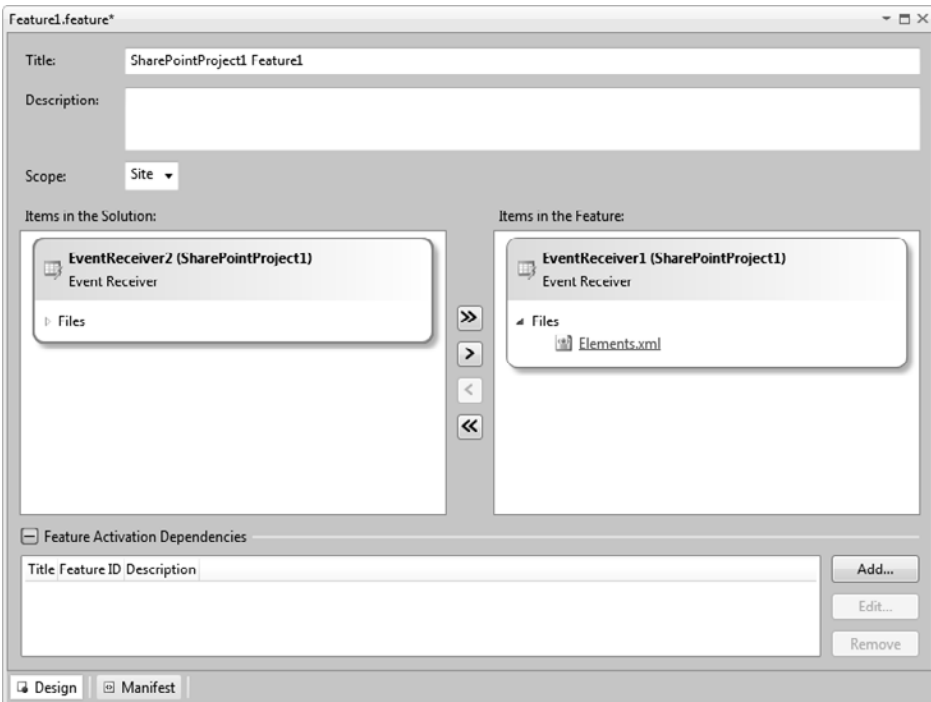


Figure 2-17: Visual Studio's Feature designer with one SharePoint project item to deploy

Feature Activation Dependencies area of the Feature designer. We will discuss these two areas of the designer next.

The Files outline shows the actual files that will be included in the feature to install the associated SharePoint project item. In this case, you can see that the Elements.xml file will be included. The assembly built with the current project is also implicitly included in the feature, even though it doesn't show in this designer.

Also, at the bottom of the dialog you can now see the Feature Activation Dependencies area of the Feature designer. Here you can add dependencies that your feature has on other features in the solution or on other features that must be installed in advance to the SharePoint site where this feature will be installed. For example, you might have a situation in which you've created two features in your solution but Feature1 needs Feature2 to be installed first. Let's enforce this constraint. Click the Add... button in the Feature Activation Dependencies area for Feature1 to specify that Feature2 is a dependency. When you click the Add... button, the dialog shown in Figure 2-18 appears. If you click the feature SharePointProject1.Feature2 and then press the Add button, Feature2 will be added to the list of Feature Activation Dependencies for Feature1.

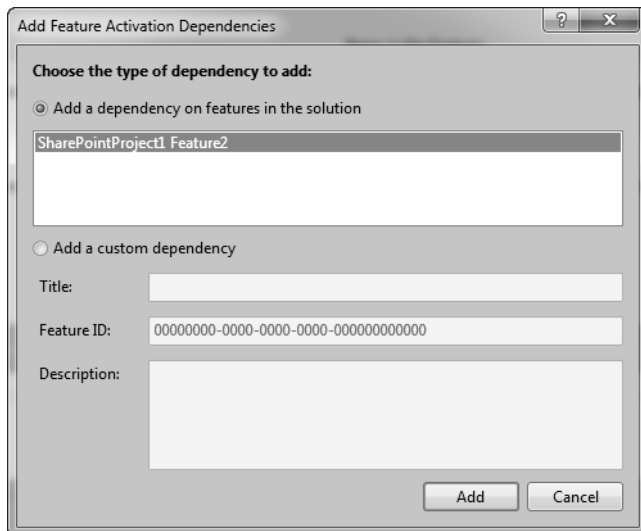


Figure 2-18: The Add Feature Activation Dependencies dialog

You also might want to add a dependency on another custom or built-in SharePoint feature. For example, you might need to ensure that the Announcement Lists feature is installed on the SharePoint site because your event receiver modifies or creates announcement lists. If announcement lists are not there, your event receiver will fail. The dialog shown in Figure 2-18 also lets you add dependencies to SharePoint features not in your solution by specifying the Feature ID of the feature on which you are dependent. As you might remember from Chapter 1, you can use the Server Explorer and the Properties window to determine the Feature ID for a particular feature as shown in Figure 1-76's DefinitionID. This ID could be added as a custom dependency for our Feature1 using the dialog in Figure 2-18.

Package Designer

Features in a project are useless unless they are deployed into what is called a Package or a .WSP file. Visual Studio helps you configure the Package created by your solution with the Package Designer. To see the Package Designer, double click on the Package.package project item under the Package folder in your solution. The designer shown in Figure 2-19 appears.

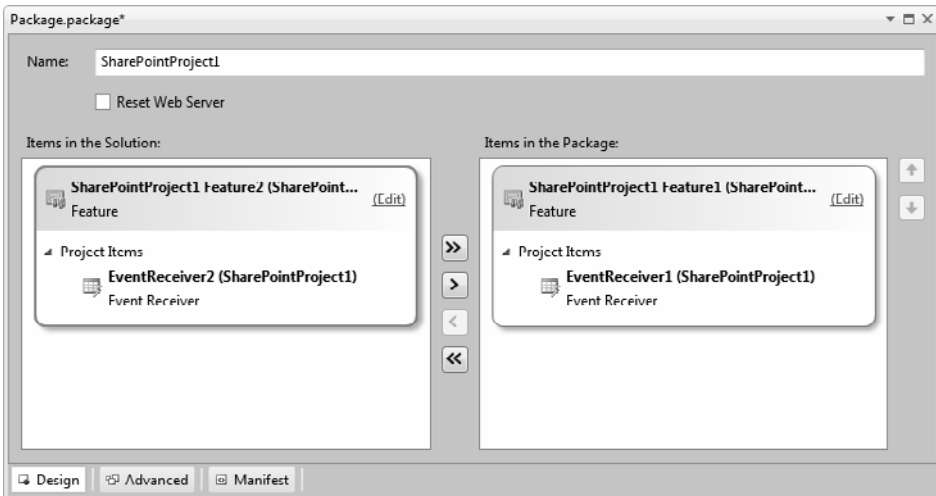


Figure 2-19: The Package Designer

When you first open the designer it won't exactly match Figure 2-19 because Visual Studio will automatically place both features we created into the items to install in the package that is created by the project. We used the < button to remove Feature2 from the package because we don't really want to install EventReceiver2 since we have no code added to it yet. Each project can build only one package, but you can have a package created by other projects in your solution. Visual Studio also lets you mix and match where features come from—that is, a feature can come from Project1 in a solution but be installed by the Package built by Project2.

If you click on the Advanced button at the bottom of the Package Designer, options are provided to add additional assemblies to the package—either assemblies created by other projects in the solution or additional external assemblies. The Advanced page of the Package Designer is shown in Figure 2-20.

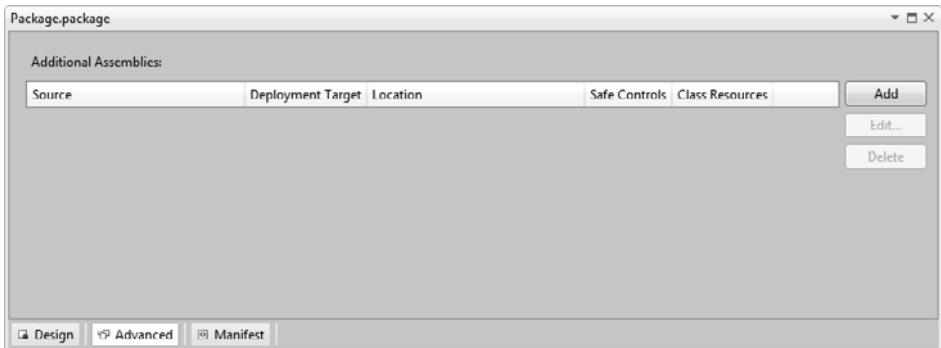


Figure 2-20: The Package Designer's Advanced Page

This has given you a brief introduction to Visual Studio's support for features and packages. We will consider features and packages in more detail in Chapter 11, "Packaging and Deployment."

Building

So we now have a SharePoint solution with two event receivers: EventReceiver1 and EventReceiver2, two features: Feature1 and Feature2, and one package: Package.package. Feature1 includes EventReceiver1, Feature2

includes EventReceiver2, but Package.package only includes Feature1, so the EventReceiver2 will not be packaged or installed. If we build currently, we will get a missing dependency error because we made Feature2 a dependency for Feature1 and Feature2 is not currently being packaged. Use the Feature designer's Feature Activation Dependencies area for Feature1 to remove the dependency on Feature2 by clicking the Remove button.

We are now ready to build our project in preparation for running and debugging it. When you build the project by choosing Build Solution from the Build menu, the Output window indicates pretty much what you would expect—it says that a DLL has been built in the Debug folder of your project called SharePointProject1.dll. When we go to the bin\debug directory for the project in Windows Explorer, you will see the DLL and the PDB for the DLL, which contains debugging information. If you package the project by choosing Package from the Build menu, you will see something a little different, as shown in Figure 2-21. You will now find in addition to the DLL and PDB files, there is a .WSP file. This is the SharePoint package file that the Feature and Package Designer helped us to create.

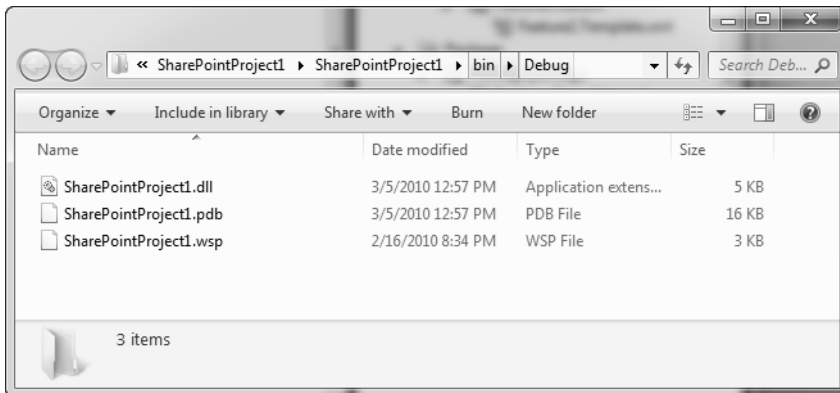


Figure 2-21: What Visual Studio built after choosing Package from the Build menu

Let's look at the .WSP file in a little more detail. Click the SharePointProject1.wsp file in the bin\debug folder of your project. Copy the SharePointProject1.wsp then Paste to make a copy of the .WSP file. Rename

its extension from .WSP to .CAB. Remember we said that a .WSP file was actually a .CAB file? Now that we've renamed it to a .CAB file, you should be able to double click on it and see the contents of the .WSP file as shown in Figure 2-22.

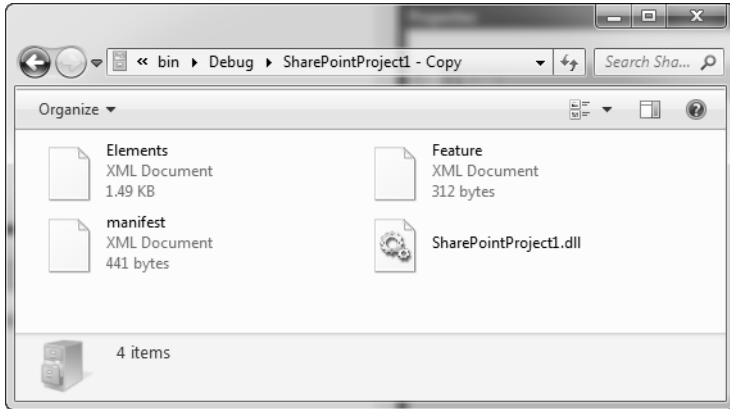


Figure 2-22: Inside the .WSP File

As you can see, there are 4 files inside the .WSP file: Elements.xml, Feature.xml, manifest.xml, and the assembly created by our project (a copy of the one we saw in the debug directory). Let's look at the contents of these files briefly. Drag Elements.xml, Feature.xml, and manifest.xml out of the renamed .CAB file to your desktop.

Manifest.xml is shown in Listing 2-4 and is the top-level manifest for the .WSP file. It tells about any assemblies included in the package (in this case SharePointProject1.dll). Additional assemblies could be included if you use the Advanced page of the Package Designer to add additional project or external assemblies. Manifest.xml also lists any features contained in the Package, in this case Feature1. You can see what manifest.xml will look like within Visual Studio by double clicking on the Package.package project item to show the Package Designer then clicking on the Manifest button at the bottom of the Package Designer. The same file shown in Listing 2-4 will be shown.

Listing 2-4: Manifest.xml inside the .WSP file

```
<?xml version="1.0" encoding="utf-8"?>
<Solution xmlns="http://schemas.microsoft.com/sharepoint/"
  SolutionId="00257823-9b84-48c4-814a-fd754b21073f"
  SharePointProductVersion="14.0">
  <Assemblies>
    <Assembly Location="SharePointProject1.dll"
      DeploymentTarget="GlobalAssemblyCache" />
  </Assemblies>
  <FeatureManifests>
    <FeatureManifest Location="SharePointProject1_Feature1\Feature.xml"
      />
  </FeatureManifests>
</Solution>
```

Feature.xml is shown in Listing 2-5 and corresponds to Feature1 of our two features. In fact, you can see this XML file by double clicking on Feature1.feature to show the Feature designer, then clicking the Manifest button at the bottom of the form. This XML file describes the Feature and tells about the manifests included in the feature. Because Feature1 includes EventReceiver1, there is one Elements.xml file associated with EventReceiver1, which is the same Elements.xml file that we found under the EventReceiver1 folder.

Listing 2-5: Feature.XML inside the .WSP file

```
<?xml version="1.0" encoding="utf-8"?>
<Feature xmlns="http://schemas.microsoft.com/sharepoint/"
  Title="SharePointProject1 Feature1" Id="d4050cd0-e7d5-48b0-88a2-fb4257b461b7"
  Scope="Site">
  <ElementManifests>
    <ElementManifest Location="EventReceiver1\Elements.xml" />
  </ElementManifests>
</Feature>
```

Elements.xml is just the same file we saw under the EventReceiver1 folder as shown in Listing 2-6. As you can see, there is no magic here, the .WSP file packages up files we've already been able to see in the Package and Feature designers and the Elements.xml file we edited in Visual Studio associated with EventReceiver1.

Listing 2-6: Elements.xml inside the .WSP File

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Receivers ListTemplateId="104">
    <Receiver>
      <Name>EventReceiver1ItemDeleting</Name>
      <Type>ItemDeleting</Type>
      <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
      <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
      <SequenceNumber>10000</SequenceNumber>
    </Receiver>
    <Receiver>
      <Name>EventReceiver1ItemAdded</Name>
      <Type>ItemAdded</Type>
      <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
      <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
      <SequenceNumber>10000</SequenceNumber>
    </Receiver>
    <Receiver>
      <Name>EventReceiver1ItemDeleted</Name>
      <Type>ItemDeleted</Type>
      <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
      <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
      <SequenceNumber>10000</SequenceNumber>
    </Receiver>
    <Receiver>
      <Name>EventReceiver1ItemAdding</Name>
      <Type>ItemAdding</Type>
      <Assembly>${SharePoint.Project.AssemblyFullName}</Assembly>
      <Class>SharePointProject1.EventReceiver1.EventReceiver1</Class>
      <SequenceNumber>1000</SequenceNumber>
    </Receiver>
  </Receivers>
</Elements>
```

Debugging

Now that we've built our project and created the .WSP file, let's debug our solution. To debug the solution, press F5 or choose Run from the Debug menu. Now we see much more activity in the Output window as shown in Listing 2-7. The Build phase does what we saw before—compiles a DLL from any code in the project and builds a package. Then in Deploy several things of interest happen. First, there are some steps to Retract the previous version of the solution. This is so the edit code, run, edit code and run again cycle will work. Visual Studio automatically removes the package

and features you installed on your last debug session before deploying your updated package to ensure that you will always have the most recent version of your solution on the server and that the old one won't conflict with the new one. You can also manually Retract a solution from the Server using the Retract command from the Build menu—for example, if you want to ensure that the Server you were testing on doesn't have your solution on it when you are done.

The next thing that Visual Studio does is deploy your .WSP file to the server—the equivalent of using stsadm.exe on the .WSP file at the command line. This installs the package, but there is also a second step after installation called activation. An installed solution is still not active for the web site. Visual Studio also activates the features in the solution to ensure they are installed and active on the web site. Visual Studio will also do an IIS Application Pool recycle if necessary—this ensures that the most current version of the site is running with your new solution installed on it. Finally, Visual Studio launches the site URL in a browser window.

Listing 2-7: Output when you start the solution

```
----- Build started: Project: SharePointProject1,
  Configuration: Debug Any CPU -----
SharePointProject1 -> C:\Users\ecarter\Documents\
Visual Studio 2010\Projects\SharePointProject1\SharePointProject1\
bin\Debug\SharePointProject1.dll
Successfully created package at:
C:\Users\ecarter\Documents\Visual Studio 2010\
Projects\SharePointProject1\SharePointProject1\
bin\Debug\SharePointProject1.wsp
----- Deploy started: Project: SharePointProject1,
  Configuration: Debug Any CPU -----
Active Deployment Configuration: Default
Run Pre-Deployment Command:
  Skipping deployment step because a pre-deployment command
  is not specified.
Recycle IIS Application Pool:
  Skipping application pool recycle because no matching package
  on the server was found.
Retract Solution:
  Skipping package retraction because no matching package
  on the server was found.
Add Solution:
  Adding solution 'SharePointProject1.wsp'...
  Deploying solution 'SharePointProject1.wsp'...
```

Activate Features:

Activating feature 'Feature1' ...

Run Post-Deployment Command:

Skipping deployment step because a post-deployment command is not specified.

=====
Build: 1 succeeded or up-to-date,

0 failed, 0 skipped =====

=====
Deploy: 1 succeeded, 0 failed, 0 skipped =====

Let's see if our event receiver is working. As you may remember, we tied our event receiver to the Announcements list. First, let's set a breakpoint in `EventReceiver1.cs`. Click on the `ItemAdded` event in that file and add a breakpoint by clicking in the left margin of the code editor. Now, go back to the web browser that Visual Studio opened up for you and navigate to the Announcements list. To get there, click on the Site Actions drop-down in the top left corner of the page and choose `View All Site Content`. On the page that appears, scroll down to the Lists section and click on the Announcements list. Click on the `Add new announcement` link at the bottom of the list as shown in Figure 2-23.

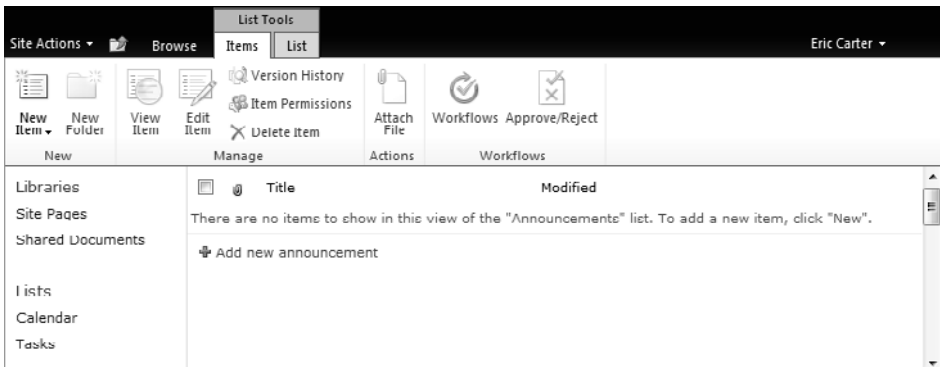


Figure 2-23: The Announcements List

In the dialog that pops up, type some text for your new announcement, something like "Test" as shown in Figure 2-24. Then click the Save button.

When you click the Save button your breakpoint should be hit in the debugger in the `ItemAdded` event. You can step through the code to watch it modify the newly added item by appending the text "added" to it using

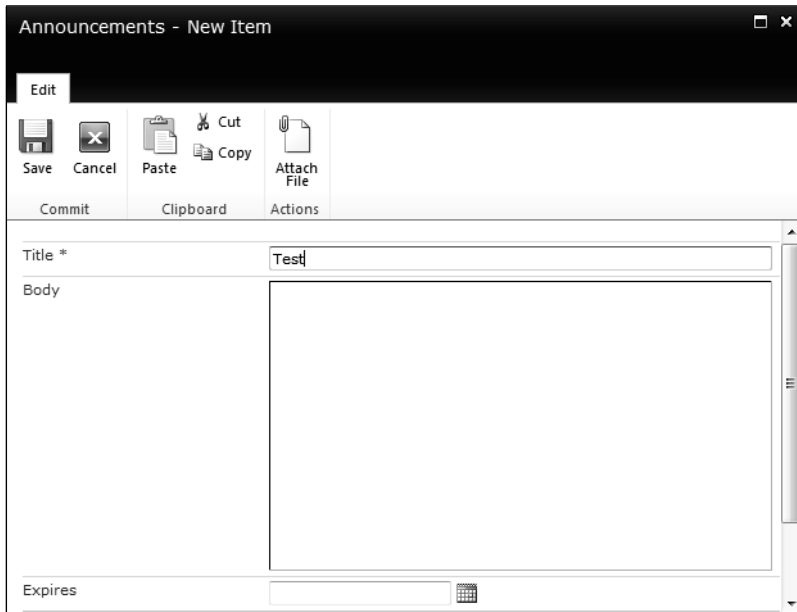


Figure 2-24: Creating a new announcement

the F10 key. Press F5 to continue. You will now see the announcement list in the browser with the newly added announcement with the text “added” appended to the text you entered “Test.”

You can also click the check box next to the newly added announcement and press the Delete Item button to try to delete it. For this case, the event receiver we created called `ItemDeleting` runs and cancels the deletion of the announcement. SharePoint shows the dialog in Figure 2-25, notifying the user that the announcement cannot be deleted.

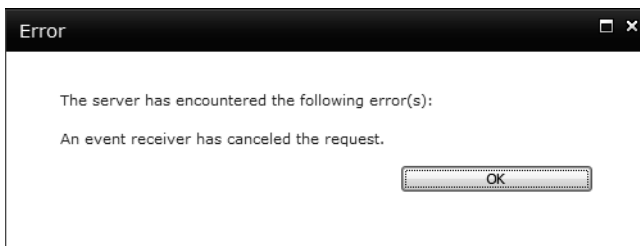


Figure 2-25: An Event Receiver canceled the Request dialog

To stop debugging, close the browser window or choose Stop Debugging from the Debug window.

Conclusion

In this chapter we have examined Visual Studio's support for SharePoint development. We have seen the SharePoint projects you can create in Visual Studio and all the project item types you can create. We have created a simple Visual Studio project with an event receiver and learned the basic structure of a Visual Studio SharePoint project in which there are SharePoint project item folders for each SharePoint project item you create that contain a code file (EventReceiver1.cs, in our example) and an Elements.xml file. We have explored the properties that are associated with SharePoint project items and their associated files. We have also seen how features and packages work in Visual Studio and examined the Feature and Package designers as well as the .WSP file created by Visual Studio. Finally, we have seen how deployment and debug work in the Visual Studio environment.

In the next chapter we will give an overview of the SharePoint object model because you will need to be familiar with the SharePoint object model to write useful SharePoint solutions.



Index

\$ (dollar sign), in tokens, 335

A

Access denied exceptions, handling, 229

Accounts

 managed, configuring, 92

 service, configuring, 92

 user, setting, 83

Activate Features step, 752

Activation dependency, features, 703

Activation properties, list workflow events,
 329

ActivationDependency, XML element, 703

ActivationProperties property, 329

Active Deployment Configuration property,
 115

Active On Default property, 727

ActivityID filter, 374

Add and Customize Pages permission level,
 66

Add Connection function, 252, 255

Add Items permission level, 65

Add Service Reference feature, 309–311

Add Solution deployment step, 752

AddContentFieldType, XML element, 705

AddItem() method, 195

Add() method, 207

Add/Remove Columns function, 252

Add/Remove Personal Web Parts permis-
 sion level, 67

Administrators for site collections, configu-
 ring, 85

Advanced property, Elements.xml file,
 129–130

Advanced Settings page, 27–28

After properties, list item events, 324

AfterProperties property, 324, 325–326

AfterUrl property, 324

Alerts

 configuring, 80

 returning, 188, 226

Alerts property

 SPUser object, 226

 SPWeb object, 188

Alerts setting, 80

Allow attachments setting, 245

Allow management of content types setting,
 246

AllowUnsafeUpdates property, 188

AlternateServerUrlFromHttpRequestUrl()
 method, 229

Always Force Install property, 727

AnimationSequence property, 627

Announcement content type, 468

Announcement lists, 14

Announcements, adding to lists, 468

Announcements, list template, 233

Anonymous access, enabling, 78, 83

Anonymous Policy command, 83

Antivirus scanning, 92

AppGlobalResource deployment type, 723,
 724–725

Application management

 accessing, 84

 Database command group, 86

 quota templates, modifying, 85

 service applications, listing, 85

 service applications, starting/stopping, 85

 Service Applications command group, 85

 Site Collections command group, 85



- Application management, site collections
 - adding, 85
 - administrators, configuring, 85
 - configuring, 85
 - deleting, 85
 - locking, 85
 - self-service site creation, enabling, 85
 - viewing, 85
- Application Page project items, 122
- Application pages
 - content placeholders, 686–690
 - Contribute permission, 693
 - creating, 683
 - customizing, 683
 - debugging, 690–692
 - default master page, specifying, 686
 - deploying, 684
 - Design permission, 693
 - design view, 684
 - DynamicMasterPageFile, 686
 - Full Control permission, 693
 - group membership permissions, 694–696
 - Limited Access permission, 693
 - overview, 682–683
 - Page directive, 686
 - permission levels, validating, 692–694
 - permissions, 692–696
 - PlaceholderAdditionalPageHead placeholder, 687
 - PlaceholderMain placeholder, 687
 - PlaceholderPageTitle placeholder, 687
 - PlaceholderPageTitleInTitleArea placeholder, 687
 - Read permission, 693
 - sample HTML markup, 688–689
 - in sandboxed solutions, 684
 - security, 692–696
 - SPSite object, disposing, 690
- Application Pages, SharePoint element, 708
- Application Pool setting, 78
- ApplicationResource deployment type, 723, 724
- Apply Themes and Borders permission
 - level, 66
- ApplyActivation workflow activity, 533
- ApplyElementManifests, XML element, 705
- Appointment list type, 364
- Appointments, scheduling, 469
- Approve Items permission level, 65
- Approve or Reject Distribution Groups
 - command, 92
- Approve() method, 214
- ASP.NET master pages, adding, 669
- AspNetHostingPermissions, 653
- ASPX files, 288
- ASPX pages, customizing, 674
- Assembly Deployment Target property, 115
- Assembly property, 138
- Assembly subelement, 132
- Associated Filter property, 393
- Association data, list workflow events, 329
- Association Editor properties, 443
- Association methods, Visual Studio external content types
 - adding, 442
 - defining, 442
 - deleting, 442
 - implementing, 444–445
 - type, 443
- Association name, 443
- AssociationData property, 329
- Associations, definition, 444
- Associations between external content types, Visual Studio
 - Association Editor properties, 443
 - association method type, 443
 - association methods
 - adding, 442
 - defining, 442
 - deleting, 442
 - implementing, 444–445
 - association name, 443
 - associations, definition, 444
 - child type descriptors, 437
 - client caching for search, 451
 - custom properties, 450–452
 - destination entity, 443
 - entity for method creation, 443
 - Finder method, creating, 434–435, 438–440
 - Finder method, enumerating roots, 451
 - foreign key association, 443
 - foreign key-based, 430
 - foreign key-less, 430
 - identifier mapping, 441–442
 - limitations, 448
 - LobSystemInstance, indexing, 451
 - model preview, 446
 - screen examples, 431–433
 - source entity, 443
 - source ID identifier, 443
 - Specific Finder method, creating, 438–440
 - type descriptor, defining, 435–438
 - type descriptor name, setting, 434–435
 - Updater method, creating, 438–440
 - updating properties, 447–448
 - user interface components, 446
- Asynchronous (post) events, 319

- Asynchronous Update function, 252
- Asynchronous updates, enabling, 252
- Attachments
 - list items, returning, 202
 - Standard View, 27
- Attachments property, 202
- Audit property
 - SPFolder object, 210
 - SPList object, 195
 - SPListItem object, 203
 - SPWeb object, 188
- Auditing
 - list items, 203
 - lists, 195
 - sites, 188
- Authentication, users, 226
- Authentication providers
 - configuring, 83
 - specifying, 78, 92
- Authentication Providers command, 83
- Authentication setting, 78
- Author, returning
 - files, 213
 - lists, 195
 - sites, 188
- Author property
 - SPFile object, 213
 - SPList object, 195
 - SPWeb object, 188
- Auto Activate In Central Admin property, 727
- Auto-retract feature, 779

B

- Backup and Restore command group, 91
- base.ItemAdding, 338
- BaseTemplate property, 196
- BaseType attribute, 275
- BaseType property, 196
- Basic Meeting Workspace subsites, 54
- Basic Page content type, 466
- Basic pages, creating, 466
- BCS (Business Connectivity Services), 361–362. *See also* External content types, SharePoint Designer; External content types, Visual Studio.
- Before (synchronous) events, 318
- Before properties, list item events, 324
- BeforeProperties property, 324, 325–326
- BeforeUrl property, 324
- Blank Meeting Workspace subsites, 54
- Blank Site subsites, 54
- Blocked File Types command, 83
- Blocking file types, 83, 92

- Blog API Settings, 80
- Blog posts, creating, 469
- Blog subsites, 54
- Bookmarks, Word documents, 514
- Books and publications
 - “Best Practices: Common Coding Issues When Using the SharePoint Object Model,” 160
 - “Best Practices: Using Disposable Windows SharePoint Services Objects,” 160
 - Visual Studio Tools for Office*, 520
- Breakpoints
 - Silverlight web parts, 634–636
 - visual web parts, 592
 - workflows, 546–547
- BreakRoleInheritance() method
 - SPList object, 196
 - SPListItem object, 203
- Browse Directories permission level, 66
- Browse User Information permission level, 66
- Browser File Handling setting, 80
- BuildAction property, Elements.xml file, 130
- Business Connectivity Services (BCS), 361–362. *See also* External content types, SharePoint Designer; External content types, Visual Studio.
- Business Data Catalog. *See* BCS (Business Connectivity Services); External content types, SharePoint Designer; External content types, Visual Studio.
- Business Data Connectivity Model project items, 122
- Business Data Connectivity Model projects, 108, 109
- Business Data Connectivity Resource Item project items, 125
- Button controls, 589

C

- Caching
 - client caching for search, 451
 - type descriptors, 394
 - Visual Studio external content types, 396
- Calendar, list template, 233
- Calendar lists, 14
- CallExternalMethod workflow activity, 530
- CAML (collaborative application markup language)
 - description, 278–279
 - getting and setting queries, 221–223
 - querying lists, 307–308
 - vs.* SPLINQ, 307–308

- Cancel property, 320
- Canceled status, events, 320
- CanExecute() method, 761–765
- CAS (Code Access Security), 647–653
- Category attribute, 603
- Central Administration page
 - Backup and Restore command group, 91
 - e-mail, configuring, 87
 - farm servers, managing, 87
 - farm-level features, managing, 87–88
 - farms, 83–84
 - Health Analyzer command group, 90
 - invoking, 74
 - monitoring server health, 88–91
 - problems and solutions, reviewing, 90
 - sandbox solutions, configuring, 87–88
 - servers, keeping up to date, 92
 - SMTP servers, configuring, 87
 - system settings, 86–88
 - timer jobs, 88–91
 - Timer Jobs command group, 90
 - Upgrade and Patch Management command group, 92
- Central Administration page, application management
 - accessing, 84
 - Database command group, 86
 - quota templates, modifying, 85
 - service applications, listing, 85
 - service applications, starting/stopping, 85
 - Service Applications command group, 85
 - site collections, 85
 - Site Collections command group, 85
- Central Administration page, Security command group
 - antivirus scanning, 92
 - Approve or Reject Distribution Groups command, 92
 - authentication providers, specifying, 92
 - Configure Information Rights Management, 93
 - Configure Managed Accounts, 92
 - Configure Password Change Settings, 92
 - Configure Self-Service Site Creation, 93
 - Configure Service Accounts, 92
 - Define Blocked File Types, 92
 - distribution groups, approving or rejecting, 92
 - farm administrators, managing, 92
 - file types, blocking, 92
 - Information Rights Management, enabling, 93
 - Manage Antivirus Settings, 92
 - Manage the Farm Administrators Group command, 92
 - Manage Web Part Security, 93
 - managed accounts, configuring, 92
 - passwords, changing, 92
 - self-service site creation, enabling, 93
 - service accounts, configuring, 92
 - Specify Authentication Providers, 92
 - web part edits, enabling, 93
 - web part security, configuring, 93
- Central Administration page, web applications
 - alerts, configuring, 80
 - Alerts setting, 80
 - anonymous access, enabling, 78, 83
 - Anonymous Policy command, 83
 - Application Pool setting, 78
 - authentication providers, configuring, 83
 - authentication providers, specifying, 78
 - Authentication Providers command, 83
 - Authentication setting, 78
 - Blocked File Types command, 83
 - Blog API Settings, 80
 - Browser File Handling setting, 80
 - change log expiration period, setting, 82
 - Change Log setting, 82
 - Claims Authentication Type setting, 78
 - Claims Based Authentication, 78
 - Classic Mode Authentication, 78
 - customer experience data, collecting, 79
 - Customer Experience Improvement Program, 79
 - Daily Time Window for Large Queries setting, 81
 - database large queries, specifying a window for, 81
 - Database Name and Authentication setting, 78
 - database operation limits, 81
 - Default Quota Template setting, 80
 - Default Time Zone setting, 80
 - Extend command, 80
 - Fallover Server setting, 79
 - file types, blocking, 83
 - Forms Based Authentication, enabling, 78, 79
 - host site, specifying, 78
 - HS Web Site setting, 78
 - HTTP Request Monitoring and Throttling, 81
 - list permissions, specifying maximum number of, 81
 - List Unique Permissions Threshold setting, 81

- List View Threshold setting, 81
- Mail Settings, 82
- making available to multiple IIS sites, 80
- Manage Features command, 82
- Managed Path command, 82
- master page default, specifying, 81
- Master Page Setting for
 - Application_Layout Pages, 81
- Maximum Upload Size setting, 81
- MetaWeblog API, enabling, 80
- monitoring HTTP requests, 81
- online presence information, enabling, 80
- overview, 76–77
- passwords, including in e-mail, 81
- path management, specifying, 82
- permission policies, setting, 83
- Permission Policy command, 83
- Presence Settings, 80
- Public URL setting, 78
- quotas for default site collections, 80
- Recycle Bin setting, 81
- root URL, specifying, 78
- RSS feeds, enabling, 80
- RSS Settings, 80
- Search Server setting, 79
- Security Configuration setting, 78
- security headers, adding to documents, 80
- security validation expiration settings, 81
- Self-Service Site Creation command, 83
- Send User Name and Password in E-Mail setting, 81
- Service Application Connections setting, 79
- Service Applications, specifying, 82
- Service Connections command, 82
- settings and policies, 77–79
- SharePoint content database, specifying, 78
- SharePoint Designer Settings, 82
- SharePoint feature management, 82
- Sign In Page URL setting, 78
- sign-in page, specifying, 78
- SSL, enabling, 78
- Text Message Service Settings, 82
- throttling HTTP requests, 81
- time zone, setting, 80
- upload limits, specifying, 81
- user accounts, setting, 83
- user names, including in e-mail, 81
- User Permissions command, 83
- User Policy command, 83
- user-created sites, enabling, 83
- User-Defined Workflows setting, 82
- Web Page Security Violation setting, 81
- web part connections, enabling, 83
- web part editing, enabling, 83
- Web Part Security command, 83
- web part updates, enabling, 83
- Windows authentication, enabling, 78
- Workflow Task Notifications setting, 82
- Change events, workflow items, 534
- Change Log setting, 82
- Check boxes, inserting, 254
- Checking file in and out, 214
- Checking workflow items in/out, 534
- CheckInItemActivity workflow activity, 534
- CheckIn() method, 214
- Checkout status, returning, 214
- CheckOutItemActivity workflow activity, 534
- Checkout() method, 214
- CheckOutType property, 214
- Child sites
 - creating, 8–12, 13
 - returning, 188
 - vs.* top-level, 57–58, 59
- Child type descriptors, 437
- Chrome State function, 255
- Chrome type, 254–255
- Chrome Type function, 254
- Circulation, creating, 467
- Circulation content type, 467
- Claims Authentication Type setting, 78
- Claims Based Authentication, 78
- Class Name property, 138
- Class subelement, 132
- Classic Mode Authentication, 78
- ClassResource deployment type, 723, 724
- Client caching for search, 451
- Client-side object model
 - creating lists programmatically, 284, 294–300
 - modifying lists programmatically, 305–307
- Close function, 255
- Close() method, 188
- Closing
 - sites, 188
 - web parts, 255
- Code Access Security (CAS), 647–653
- Code View *vs.* Source View, 589
- Code workflow activity, 530
- Code-only web parts. *See* Web parts, code-only.
- Collaborative application markup language (CAML). *See* CAML (collaborative application markup language).

- Colors, site settings, 70
- Columns
 - adding/removing, custom list views, 252
 - in documents, 39–40
 - in list views, 19, 21
 - lists. *See* List columns.
 - totals views, 40
 - in Word documents, 509
- Columns, Standard View
 - arranging, 19
 - Create Column dialog, 23, 24
 - custom site columns, 27
 - List tab, 22
 - reusing column definitions, 22, 24, 26
 - from scratch, 22
 - selecting, 19
 - settings, illustration, 21
- Columns setting, 39
- Commands, site collection administration, 74. *See also specific commands.*
- Comment content type, 468
- Comments, creating, 468
- Comparison filter, 374
- CompensatableSequence workflow activity, 531
- CompensatableTransaction workflow activity, 531
- Compensate workflow activity, 531
- Compensation-related workflow activities, 531
- CompleteTask workflow activity, 533
- Completion type, list workflow events, 329
- CompletionType property, 329
- Conditional formatting, custom list views, 251
- Conditional Formatting function, 251
- ConditionedActivityGroup workflow activity, 530
- Configuration property, 187
- Configure Information Rights Management, 93
- Configure Managed Accounts, 92
- Configure Password Change Settings, 92
- Configure Self-Service Site Creation, 93
- Configure Service Accounts, 92
- Connecting web parts, 255
- Connection properties, custom list views, 252
- Connection strings, web part configuration, 643–647
- Connections, adding to custom list views, 252
- Connections function, 255
- Consumers, web part connections, 637
- Contact content type, 468
- Contact information, storing, 468
- Contact list type, 364
- Contact lists, 14
- Contacts, list template, 233
- Container workflow activities, 529
- Content. *See* Libraries; Lists; Pages; Subsites (workspaces); Workflows.
- Content approval, files, 214
- Content controls
 - adding to documents, 500–504
 - renaming, 514
- Content Editor web parts, 50
- Content placeholders, application pages, 686–690
- Content Type, SharePoint element
 - associated project items, 707, 709
 - conflict resolution, 776
- Content Type Binding, SharePoint element, 707, 709
- Content type groups, Document content types
 - Basic Page, 466
 - Document, 466
 - Dublin Core Columns, 466
 - Form, 466
 - Link to a Document, 466
 - List View Style, 466
 - Master Page, 466
 - Picture, 466
 - Web Part Page, 466
 - Wiki Page, 466
- Content type groups, Folder content types
 - Discussion, 467
 - Folder, 467
 - Summary Task, 467
- Content type groups, Group Work content types
 - Circulation, 467
 - Holiday, 467
 - New Word, 467
 - Official Notice, 468
 - Phone Call Memo, 468
 - Resource, 468
 - Resource Group, 468
 - Timecard, 468
 - Users, 468
 - What's New Notification, 468
- Content type groups, List content types
 - Announcement, 468
 - Comment, 468
 - Contact, 468
 - East Asia Contact, 468
 - Event, 469

- Issue, 469
- Item, 469
- Link, 469
- Message, 469
- Post, 469
- Reservations, 469
- Schedule, 469
- Schedule and Reservations, 469
- Task, 469
- Content type groups, special content types, 469
- Content type hierarchy
 - diagram, 465
 - inheritance, 464
 - site content types, 464
- Content Type projects, 108, 109, 122–123
- Content types
 - associating with workflows, 521–522
 - associations, workflows, 548–552
 - default, setting, 499–500
 - definition, 458–459
 - displaying, 491–492
 - documents, managing, 43–46
 - libraries, managing, 43–46
 - list columns, 459
 - list items, returning, 203
 - lists. *See* Lists, content types.
 - overview, 457–458
 - properties, displaying, 95–97
 - Standard View, 27
- Content types, site columns
 - additional settings, 461
 - column data types, 460
 - column group name, 460–461
 - column name, 460
 - components of, 459–461
 - creating, 461–464
 - predefined column groups, 461
- Content types, Visual Studio
 - adding fields, 476–477
 - content type ID, specifying, 473, 482
 - content type name, 473
 - content type version, 473
 - ContentType element attributes, 473
 - creating, 470–472
 - deploying the project, 473–477
 - description, adding, 473
 - event handler, creating, 483–487
 - feature dependency, 483–487
 - fields, adding, 476–477
 - field/site column inheritance, 473
 - group name, specifying, 473
 - ID, specifying, 473, 482
 - list definition, creating, 487–491
 - list template types, 489–490
 - name, specifying, 473
 - picture library, creating, 476–480
 - removing fields, 474–476
 - site columns, creating, 481–483
 - version, 473
- Content types, Word documents
 - adding to a library, 497–499
 - associating documents with content types, 507
 - basic project, creating, 492–494
 - bookmarks, 514
 - changing default view, 507–508
 - columns in view, changing, 509
 - content controls, adding, 500–504
 - content controls, renaming, 514
 - content types, displaying, 491–492
 - deploying the project, 506–509, 517–520
 - document library, creating, 497
 - Document Panel feature, 491–492
 - mapped folders, 507
 - overview, 491
 - setting a default content type, 499–500
 - site columns, adding to default library view, 504–506
 - VSTO (Visual Studio Tools for Office), 512–520
 - Word documents, creating, 495–496, 511–517
 - Word template, creating, 494–495
 - Word template, uploading, 495–497
- Content Types SharePoint element, sand-boxed solutions, 778
- ContentDatabases property, 180
- ContentType element attributes, 473
- ContentType property, 203
- ContentTypes property, 196
- Context information
 - getting. *See* SPContext object.
 - object models, 154
- Context property, 324
- Contribute permission, 28–29, 693
- Control flow, workflow activities, 529
- Controls
 - adding, visual web parts, 587–589
 - registering, site pages, 680
- Convert() method, 213
- Copy to Output Directory property, Elements.xml file, 130
- Copying
 - files in workflows, 535
 - list items, 202, 203
 - workflow items, 535
- CopyItemActivity workflow activity, 535

- CopyTo() method
 - SPFolder object, 211
 - SPListItem object, 203
 - Correlation tokens, 544
 - Count property, 207
 - Counting
 - folders, list items, 207
 - folders, lists, 196
 - list items, 196, 207
 - Create, Read, Update, Delete (CRUD) methods. *See* CRUD (Create, Read, Update, Delete) methods.
 - Create a version... setting, 246
 - Create Alerts permission level, 65
 - Create Column dialog, 23, 24
 - Create Groups permission level, 66
 - Create method, creating, 417–420
 - Create Subsites permission level, 66
 - CreateChildControls method, 167, 173–174
 - Created property
 - SPList object, 194
 - SPWeb object, 187
 - CreateISO8601DateTimeFromSystemDateTime() method, 228
 - CreateItemActivity workflow activity, 535
 - Create() method, 177
 - CreateTask workflow activity, 533
 - CreateTaskWithContentType workflow activity, 533
 - Creator Field property, 393
 - Creator method instance type, 395
 - Creator property, 727
 - CRUD (Create, Read, Update, Delete) methods, 368–375
 - CRUD-enabling, Visual Studio external content types
 - accessing connection information at runtime, 421–425
 - code management, 426
 - code sample, 426–428
 - connection information, storing, 421–425
 - Create method, creating, 417–420
 - creating an empty project, 401–407
 - default methods, renaming, 408–409
 - Delete method, creating, 417–420
 - deploying the solution, 428–429
 - entity shapes, renaming, 407–408
 - identifier, renaming, 408
 - LINQ to SQL, 403
 - methods, implementing, 426–428
 - testing the solution, 428–429
 - type descriptor, copying, 414–417
 - type descriptor, defining, 409–414
 - type descriptor, manually adjusting, 421
 - Type Name property, setting, 408
 - Update method, creating, 417–420
 - Currency (money) settings, regional, 189
 - Current property, 172
 - CurrentUser property, 189
 - CurrentUserid property, 324
 - Custom Action, SharePoint element, 707, 709
 - Custom Action Group, SharePoint element, 707, 709
 - Custom Actions, 778
 - Custom List, list template, 234
 - Custom List in Datasheet View, list template, 234
 - Custom list views. *See* List views, custom.
 - Custom lists, 14, 16
 - Custom properties
 - associations between external content types, 451
 - method instance, Visual Studio external content types, 396
 - RootFinder, 451
 - ShowInSearchUI, 451
 - UseClientCachingForSearch, 451
 - Visual Studio external content types, 450–452
 - Custom Properties property
 - method instance, Visual Studio external content types, 396
 - type descriptors, Visual Studio external content types, 393
 - Custom Tool Namespace property, Elements.xml file, 130
 - Custom Tool property, Elements.xml file, 130
 - Customer experience data, collecting, 79
 - Customer Experience Improvement Program, 79
 - Customization scope, setting for visual web parts, 603
 - Customize XSLT function, 254
 - Customizing
 - application pages, 683
 - master pages, 661–671
 - site pages, 673–682
 - CustomUpgradeAction, XML element, 705
- D**
- Daily Time Window for Large Queries setting, 81
 - Data Elements properties, 370. *See also specific properties.*
 - Data source details, displaying, 252
 - Data Source Details function, 252

- Data Source Elements
 - display name, 370
 - displaying in the picker, 370
 - fields, 370
 - foreign key identifier, 370
 - identifiers, listing, 370
 - .NET Type, 370
 - Office Item type associations, 370
 - read-only, 370
 - required, 370
 - time stamp of last update, 370
 - unique identifier, selecting, 370
- Data view controls, inserting, 254
- Data View Preview function, 254
- Database command group, 86
- Database Name and Authentication setting, 78
- Databases
 - displaying data from, 643–647
 - large queries, specifying a window for, 81
 - operation limits, overriding, 81
 - operation limits, specifying, 81
 - SharePoint content, specifying, 78
- Data-driven workflows, creating, 530
- Datasheet View, 15, 16, 18, 28
- DataSource property, 196
- DataTable objects, from list items, 207
- Date and time
 - formatting, 229
 - parsing, 229
 - representing in ISO8061 format, 228
- Deactivating features, 712
- Deadlines, creating, 469
- Debug assemblies, empty projects, 114
- Debugging
 - application pages, 690–692
 - Developer Dashboard, 802–815
 - diagnostic tools for, 802–815
 - event receivers, 347–358
 - Silverlight, 779–786
 - Silverlight web parts, 634–636
 - visual web parts, 592–597
 - workflows, 562
- Decision Meeting Workspace subsites, 54
- Decision points, workflow activities, 529
- Declarative workflows, 562
- Default Display Name property
 - method instance, Visual Studio external content types, 396
 - type descriptors, Visual Studio external content types, 393
- Default finder method, custom list views, 251
- Default page, custom list views, 248
- Default property, 396
- Default Quota Template setting, 80
- Default Resource File property, 727
- Default Time Zone setting, 80
- DefaultItemOpen property, 196
- DefaultValue property, 218
- DefaultView property, 196
- DefaultViewUrl property, 197
- Define Blocked File Types, 92
- Definition property, 331
- Delay workflow activity, 530
- Delaying workflows, 530
- Delegate Controls, SharePoint element, 707, 709
- Delete Items permission level, 65
- Delete method, creating, 417–420
- Delete This Site action, 72
- Delete Versions permission level, 65
- Deleted items, restoring, 74
- DeleteItemActivity workflow activity, 535
- DeleteItemById() method, 208
- Delete() method
 - SPField object, 217
 - SPFile object, 213
 - SPFolder object, 210
 - SPList object, 195
 - SPListItem object, 202
 - SPListItemCollection object, 207
 - SPWeb object, 187
 - SPWebApplication object, 179
- Deleter method instance type, 395
- DeleteTask workflow activity, 533
- Deleting
 - association methods, 442
 - columns, lists, 217
 - entity objects, Visual Studio external content types, 395
 - fields, lists, 217
 - files, 213
 - folders, list items, 207, 208
 - list items, 202, 207–208
 - lists, 28, 195
 - methods, Visual Studio external content types, 389
 - site collections, 85
 - sites, 187
 - tasks, 533
 - web applications, 179
 - workflow items, 535
- Deletion events, workflow items, 534
- Dependent assemblies, 138
- DependentValuePath property, 627

- Deploying. *See also* Retracting.
 - application pages, 684
 - changes only, 452–454
 - custom list views, 255–261
 - deplysolutions command, 720
 - event receivers, 347–358
 - external content types, Visual Studio, 385–386, 452–454
 - features, 709–712
 - full, 452–454
 - importing to a feature event receiver, 454
 - incremental, 452–454
 - list instances, 283
 - master pages, 671
 - projects. *See* Projects, deployment.
 - Silverlight web parts, 630–636
 - site pages, 682
 - solutions, 717–720
 - Visual Studio external content types, 428–429
 - Visual Studio projects. *See* Visual Studio projects, configurable deployment; Visual Studio projects, deployment.
 - visual web parts as farm solutions, 583
 - workflows, 546–548
- Deployment Location property, 130, 722–725
- Deployment Path property, 727
- Deployment Server Type property, 746
- Deployment Type property, 722–725
- Deployment Type property, Elements.xml file, 130
- Description, adding to content types, 473
- Description attribute, 275, 279, 473
- Description property
 - SPField object, 217
 - SPList object, 194
 - SPWeb object, 187
 - Visual Studio feature designer, 727
 - Visual Studio projects, package designer, 746
- Design permission, application pages, 693
- Design tab functions, 253–254
- Design view, application pages, 684
- Destination entity, 443
- Developer Dashboard, 802–815
- Diagnostic tools for debugging, 802–815
- Discussion Board, list template, 234
- Discussion board lists, 15
- Discussion content type, 467
- Display name
 - Data Source Elements, 370
 - default, external content types, 396
- Display Name property, 370
- Display New Folder command... setting, 245
- DisplayName attribute, 275
- DisplayName property
 - SPListItem object, 203
 - SPWebApplication object, 180
- Display...on the Quick Launch setting, 245
- DisplaySize property, 218
- Dispose Checker tool, 160
- Dispose() method, 157–158
- Disposing of managed objects. *See also* Garbage collection.
 - coding patterns, 157–160
 - Dispose Checker tool, 160
 - Dispose method, 157–158
 - IDisposable objects, 157–160
 - object models, 155
 - SPRequest, 155
 - SPSite, 157–158
 - SPSite object, 690
 - SPWeb, 157
 - try, catch, finally pattern, 159
 - using keyword, 158
- Distribution groups, approving or rejecting, 92
- Document content types, 466. *See also specific content types.*
- Document Converter, SharePoint element, 707, 709
- Document libraries
 - creating, 497
 - folders, 155
- Document Libraries subnode, 98
- Document Library, list template, 234
- Document Panel feature, 491–492
- Document Workspace subsites, 54
- Documents
 - basic pages, creating, 466
 - checking in and out, 37
 - columns, displaying and arranging, 37, 40
 - content types, managing, 43–46
 - copying, 37
 - creating, 466
 - downloading, 37
 - Dublin Core metadata, 466
 - editing, 37
 - entering data into forms, 466
 - linking between, 466
 - list view styles, creating, 466
 - managing. *See* Libraries, document.
 - master pages, creating, 466
 - pictures in picture libraries, 466
 - revision history, 37
 - versioning, 35

- views, 37–41
- web part hosts, 466
- wiki pages, creating, 466
- Documents, view settings
 - columns, 39
 - filter, 39
 - folders, 40
 - group by, 40
 - inline editing, 40
 - item limit, 40
 - mobile, 40
 - sort, 39
 - style, 40
 - tabular view, 40
 - totals, 40
- Documents ribbon tab, 37
- Dollar sign (\$), in tokens, 335
- Dublin Core Columns content type, 466
- Dublin Core metadata, 466
- DwpFile deployment type, 723
- Dynamic web part connections, 637
- DynamicMasterPageFile, 686
- E**
- East Asia Contact content type, 468
- Edit in personal view function, 255
- Edit Items permission level, 65
- Edit permission, 380
- Edit Personal Information permission level, 67
- Edit View page, 39
- Editing
 - Content Editor, 50
 - datasheet view, enabling, 28
 - documents, 37, 40
 - Elements.xml file, 276
 - inline, 40
 - list items, 31–33
 - permission levels, 63–64
 - visual web parts, 605
 - web parts, 50, 83, 93, 255
- Editing view, visual web parts, 603, 604–605
- Element descriptions, features, 705
- Element Manifest file. *See* Elements.xml file.
- ElementFile, XML element, 703
- ElementManifest, XML element, 701–706
- ElementManifest deployment type, 723
- ElementManifests, XML element, 701–705
- Elements. *See* SharePoint elements; XML elements.
- Elements.xml file. *See also* List definitions, creating with Visual Studio.
 - Assembly subelement, 132
 - attributes, 275, 279
 - BaseType attribute, 275
 - Class subelement, 132
 - code sample, 131
 - definition, 274
 - Description attribute, 275, 279
 - DisplayName attribute, 275
 - editing, 276
 - ElementManifests element, 701–702
 - Image attribute, 275
 - Name attribute, 275
 - Name subelement, 132
 - OnQuickLaunch attribute, 275, 279
 - overview, 128–129
 - SecurityBits attribute, 275
 - Sequence attribute, 275
 - Sequence Number subelement, 132
 - subelements, 132
 - TemplateType attribute, 279
 - Title attribute, 279
 - Type attribute, 275
 - Type subelement, 132
 - Url attribute, 279
 - visual web parts, 583
 - workflows, 542
- Elements.xml file, properties
 - Advanced, 129–130
 - BuildAction, 130
 - Copy to Output Directory, 130
 - Custom Tool, 130
 - Custom Tool Namespace, 130
 - Deployment Location, 130
 - Deployment Type, 130
 - File Name, 130
 - Full Path, 130
 - Misc, 129–130
 - SharePoint, 129–130
- E-mail
 - address, getting and setting, 226
 - configuring, 87
 - events, 327
 - sending, 228, 534
- Email property, 226
- Empty Element project items, 123
- Empty elements, 731
- Empty projects. *See* Project types, empty.
- Empty SharePoint Project, 107
- EnableWorkflowModification workflow activity, 533
- EnsureListItemsData() method, 208
- EnsureUser() method, 189
- Entities
 - classes, generating, 308–311
 - external content types, 389
 - for method creation, 443

- Entities, *continued*
 - objects, Visual Studio external content types, 395
 - shapes, renaming, 407–408
- Enumerate Permissions permission level, 66
- Error messages
 - from events, setting, 320
 - from exceptions, returning, 228
 - external content types, SharePoint Designer, 371, 374–375
- ErrorCode property, 228
- ErrorException property, 329
- ErrorMessage property, 320
- Event content type, 469
- Event handlers
 - content types, creating, 483–487
 - workflows, setting, 545
- Event Receiver, SharePoint element, 707, 709
- Event Receiver projects, 108, 109, 123
- Event receiver properties
 - Cancel, 320
 - ErrorMessage, 320
 - EventType, 320
 - Receiver Assembly, 345
 - Receiver Class, 345
 - ReceiverData, 320
 - RedirectUrl, 320
 - SiteId, 320
 - Status, 320
 - Web, 320
- Event Receivers, 778
- Event receivers. *See also* Events.
 - debugging, 347–358
 - Definition property, 331
 - deploying, 347–358
 - Elements Manifest file. *See* Elements.xml file.
 - feature events, 330–331, 341–347
 - Feature property, 331
 - GUID, changing, 345
 - list, 126
 - list e-mail, 126
 - list item, 126
 - overview, 317–318
 - root item node or folder. *See* EventReceiver1 folder.
 - SPContext object, 175–176
 - SPEmailEventReceiver, 327
 - SPFeatureReceiverProperties object, 331
 - SPItemEventProperties object, 324–325, 325–326
 - SPItemEventReceiver, 322–327
 - SPListEventProperties object, 321–322
 - SPListEventReceiver, 321–322
 - SPWebEventProperties object, 328
 - SPWebEventReceiver, 327–328
 - SPWorkflowEventProperties object, 329–330
 - SPWorkflowEventReceiver, 328–330
 - testing, 150
 - types of, 126
 - UserCodeSite property, 331
 - web, 126
- Event receivers, code file (EventReceiver1.cs)
 - code sample, 133–134, 134–135
 - overview, 133
- Event receivers, creating in Visual Studio
 - adding/removing event handlers, 336–338
 - base.ItemAdding, 338
 - in C#, 331–336
 - error messages, 338–341
 - farm solutions, 332
 - sandbox solutions, 332
 - tokens, 335
- Event receivers, returning
 - files, 214
 - lists, 196
- EventDriven workflow activity, 530
- Event-driven workflows, grouping, 530
- EventHandlingScope workflow activity, 530
- EventReceiver1 folder
 - dependent assemblies, 138
 - events, 138. *See also specific events.*
 - overview, 136
 - Project Output References, 138
 - properties, 137–139. *See also specific properties.*
 - Safe Control Entries, 138
 - trusted entities, identifying, 138
- EventReceiver1.cs file
 - code sample, 133–134, 134–135
 - overview, 133
- EventReceivers property
 - SPFile object, 214
 - SPList object, 196
- Events. *See also* Event receivers.
 - asynchronous (post), 319
 - canceled status, determining, 320
 - current SPWeb, 320
 - e-mail, 327
 - error messages, setting, 320
 - event type, determining, 320
 - external, listening to, 530
 - for list items. *See* List item events.
 - for list workflow. *See* List workflow events.

- for lists. *See* List events.
 - receiver data, 320
 - redirect URL, 320
 - site GUID, 320
 - SPEmailEventReceiver, 327
 - status, 320
 - synchronous (before), 318
 - task changes, 533
 - triggers for, 319–320
 - type, determining, 320
 - workflow activation, 534
 - workflow item changes, 534
 - workflow item deletion, 534
- Events (calendar), creating, 469
- EventType property, 320
- Exception handling, workflow activities, 530
- Exceptions. *See also* SPException object.
- access denied, handling, 229
 - error messages, returning, 228
 - identification, returning, 228
 - information, list workflow events, 329
 - raising, workflow activities, 532
 - stack traces, returning, 228
- Execute permission, 380
- Execute() method, 761–765
- Exists property
- SPFile object, 213
 - SPFolder object, 210
 - SPWeb object, 188
- Extend command, 80
- External content types, SharePoint Designer
- Appointment list type, 364
 - Contact list type, 364
 - CRUD (Create, Read, Update, Delete) methods, 368–375
 - Data Elements properties, 370. *See also specific properties.*
 - data retrieval criteria, 376–377
 - Data Source Elements
 - display name, 370
 - displaying in the picker, 370
 - fields, 370
 - foreign key identifier, 370
 - identifiers, listing, 370
 - .NET Type, 370
 - Office Item type associations, 370
 - read-only, 370
 - required, 370
 - time stamp of last update, 370
 - unique identifier, selecting, 370
- Edit permission, 380
- errors and warnings, 371
- Execute permission, 380
- external lists, 377
- external source, connecting to, 362–366
- external source types, 366
- filters
- ActivityID, 374
 - Comparison, 374
 - creating, 371–373
 - errors and warnings, 374–375
 - Limit, 374
 - Page Number, 374
 - Timestamp, 374
 - types of, 374
 - Wildcard, 374
- finder methods, 376
- Generic List type, 364
- .NET external source types, 366
- Office Item Types, 364
- overview, 362–364
- permissions for external content, 378–380
- Post list type, 364
- Secure Store Service, 367
- security settings, 380–381
- Selectable in Clients permission, 381
- Set Permissions permission, 381
- SQL Server connection, 367–368
- SQL Server external source types, 366
- Task list type, 364
- user interface, creating, 376–380
- viewing and managing external settings, 364
- viewing results, 382–384
- vs.* Visual Studio, 384–385
- WCF Service external source types, 366
- External content types, Visual Studio
- adding entities and associations, 389
 - associations between external content types
 - Association Editor properties, 443
 - association method type, 443
 - association methods, adding, 442
 - association methods, defining, 442
 - association methods, deleting, 442
 - association methods, implementing, 444–445
 - association name, 443
 - associations, definition, 444
 - child type descriptors, 437
 - client caching for search, 451
 - custom properties, 450–452
 - destination entity, 443
 - entity for method creation, 443
 - Finder method, creating, 434–435, 438–440
 - Finder method, enumerating roots, 451
 - foreign key association, 443

- External content types, Visual Studio
 - associations between external content types, *continued*
 - foreign key-based, 430
 - foreign key-less, 430
 - identifier mapping, 441–442
 - limitations, 448
 - LobSystemInstance, indexing, 451
 - model preview, 446
 - screen examples, 431–433
 - source entity, 443
 - source ID identifier, 443
 - Specific Finder method, creating, 438–440
 - type descriptor, defining, 435–438
 - type descriptor name, setting, 434–435
 - Updater method, creating, 438–440
 - updating properties, 447–448
 - user interface components, 446
 - external content types, Visual Studio
 - caching, 396
 - components available, 388
 - creating methods, 389
 - CRUD-enabling
 - accessing connection information at runtime, 421–425
 - code management, 426
 - code sample, 426–428
 - connection information, storing, 421–425
 - Create method, creating, 417–420
 - creating an empty project, 401–407
 - default methods, renaming, 408–409
 - Delete method, creating, 417–420
 - deploying the solution, 428–429
 - entity shapes, renaming, 407–408
 - identifier, renaming, 408
 - LINQ to SQL, 403
 - methods, implementing, 426–428
 - testing the solution, 428–429
 - type descriptor, copying, 414–417
 - type descriptor, defining, 409–414
 - type descriptor, manually adjusting, 421
 - Type Name property, setting, 408
 - Update method, creating, 417–420
 - deleting methods, 389
 - deployment, 452–454
 - deployment options, 385–386
 - display name, default, 396
 - displaying method definitions, 389
 - entity objects, 395
 - filter descriptors, 389
 - getting started, 386–388
 - importing BDC models from SharePoint Designer, 385
 - method instance, 396
 - method instance properties, 396. *See also specific properties.*
 - method instance types, 395
 - method parameter directions, 389–390
 - method signatures, 395
 - overview, 384–386
 - reviewing code, 390–392
 - vs. SharePoint Designer, 384–385
 - type descriptor properties, 393–394. *See also specific properties.*
 - type descriptors
 - caching, 394
 - creating a new ECT, 393
 - custom properties, 393
 - data structure name, 394
 - display name, default, 393
 - filters, setting, 393
 - foreign identifier association entity, 393
 - foreign identifier association name, 393
 - identifier, 393
 - identifier entity, 393
 - name, 394
 - .NET type, 394
 - read-only, 394
 - representing collections, 394
 - significant values, 394
 - Update Field, 394
 - updating, 394
 - External hosts, getting/sending data, 530
 - External lists, 15, 377
 - External source, connecting to, 362–366
 - External source types
 - .NET, 366
 - SQL Server, 366
 - WCF Service, 366
- F**
- Fallover Server setting, 79
 - Farm administrators, managing, 92
 - Farm scope, features, 700–701
 - Farm servers, managing, 87
 - Farm solutions
 - creating, 163
 - ItemAdding event handler, 166
 - List Item Event handler, adding, 163–166
 - sample code, 164
 - Visual Studio configurable deployment, 777–778
 - Visual Web Part, adding, 163–164
 - Farm-level features, managing, 87–88
 - Farms. *See also* SPFarm object.
 - available services, determining, 177
 - available solutions, returning, 177

- creating, 177
- definition, 83–84
- GUIDs, returning, 177
- local computers, adding, 177
- opening, 177
- physical computers, returning, 177
- SPFarm object, returning, 177
- SPServerCollection object, returning, 177
- SPServiceCollection object, returning, 177
- SPSolutionCollection object, returning, 177
- updating, 177
- FaultHandler workflow activity, 530
- Faults, sending to the web service, 532
- Feature designer. *See* Visual Studio projects, feature designer.
- Feature events, event receivers, 330–331, 341–347
- Feature Id property, 727
- Feature manifest file. *See also* Feature.xml file.
 - definition, 700
 - ElementManifests element, 701–702
 - example, 702–703
- Feature Properties property, 137
- Feature property, 331
- Feature Receiver property, 138
- FeatureActivated event, 138
- FeatureDeactivating event, 138
- FeatureId property, 321
- FeatureInstalled event, 138
- Features
 - activation dependency, 703, 739–743
 - deactivation, 712
 - definition, 71–72
 - dependency, 483–487
 - deploying, 709–712
 - displaying, 98, 99
 - Farm scope, 700–701
 - farm-level, managing, 87–88
 - IDs, 700
 - installing, 710–711
 - installing *vs.* activating, 346
 - managing, 71–72
 - manual activation, 711
 - names, 700
 - overview, 699–706
 - properties, defining, 704
 - scope, 700–701
 - Site scope, 700–701
 - supporting files, specifying, 703
 - uninstalling, 711
 - Web Application scope, 700–701
 - Web scope, 700–701
- Features, upgrade actions
 - element descriptions, 705
 - fields, adding, 705
 - files, moving or renaming, 705
 - overview, 704
 - parameters, specifying, 705
 - provisioning elements, 705
- Features folder, 541–542
- Features property
 - SPWeb object, 187
 - SPWebApplication object, 179
- Features subnode, 98, 99
- Feature/Site Template Association, SharePoint element, 707, 709
- FeatureUninstalling event, 138
- Feature.xml file, XML elements
 - ActivationDependency, 703
 - AddContentFieldType, 705
 - ApplyElementManifests, 705
 - code sample, 702
 - CustomUpgradeAction, 705
 - descriptions, 703–704
 - ElementFile, 703
 - ElementManifest, 701–706
 - ElementManifests, 701–705
 - MapFile, 705
 - Parameter, 705
 - pointing to, 704
 - Property, 704
 - UpgradeActions, 704–706
 - VersionRange, 705
- Field Element, SharePoint element, 707, 709
- Field property
 - Data Elements, 370
 - SPListEventProperties object, 321
- FieldName property, 321
- FieldRenderingControl property, 218
- Fields. *See also* Site columns.
 - adding to content types, 476–477
 - Data Source Elements, 370
 - features, adding, 705
 - vs.* list columns, 235
 - removing from content types, 474–476
- Fields, lists. *See also* SPField object.
 - default value, getting and setting, 218
 - deleting, 217
 - deriving other fields from, 218
 - description, returning, 217
 - display name, returning, 218
 - display size, returning, 218
 - edit form, displaying, 218
 - group of site columns, returning, 217
 - GUID, returning, 217
 - internal name, getting and setting, 218

Fields, lists, *continued*

- internal name, returning, 217
- name, returning, 217
- parent list, returning, 218
- rendering, 218
- required, setting, 217
- schema XML, getting and setting, 218
- scope, returning, 218
- sealing, 218
- site-relative path, returning, 218
- SPMobileBaseFieldControl class, returning, 218
- title, returning, 217
- type, returning, 217
- updating, 217
- version, returning, 217
- visibility, setting, 217

Fields property

- SPContext object, 173
- SPList object, 194
- SPListItem object, 202
- SPListItemCollection object, 207

Fields SharePoint element, sandboxed solutions, 778

Field/site column inheritance, 473

FieldXml property, 322

File Name property, Elements.xml file, 130

File paths, 353–354

File property

- SPContext object, 173
- SPListItem object, 202

File types, blocking, 83, 92

Files. *See also* SPFile object.

- author, returning, 213
- checking in, 214
- checking out, 214
- checkout status, returning, 214
- content approval, 214
- copying, 535
- creating, 535
- deleting, 213
- document library, determining, 214
- event receivers, returning, 214
- existence, determining, 213
- features, moving or renaming, 705
- format conversion, 213
- level, returning, 213
- managing. *See* Libraries.
- moving, 214
- name, returning, 213
- opening as binary, 213
- properties, returning, 213
- publishing method, returning, 213
- recycling, 214
- server-relative URL, returning, 214

- site, returning, 213
- site-relative URL, returning, 213
- size, returning, 213
- SPLimitedWebPartManager object, returning, 214
- SPListItem object, returning, 213
- title, returning, 213
- updating, 213
- version history, returning, 213
- in workflows, 535, 542–543

Files property

- SPFolder object, 210
- SPWeb object, 187

FileSystemObjectTypePropertyName property, 324

Filter descriptors, external content types, 389

Filter function, 251

Filter setting, 39

Filters

- ActivityID, 374
- Comparison, 374
- creating, 371–373
- custom list views, 251
- document views, 39
- errors and warnings, 374–375
- Limit, 374
- Page Number, 374
- Timestamp, 374
- types of, 374
- Wildcard, 374

Finder function, 251

Finder method

- creating, 434–435, 438–440
- enumerating roots, 451

Finder method instance type, 395

Finder methods, 376

Finding entity objects, Visual Studio external content types, 395

Folder content types, 467

Folder Name property, 137

Folder property

- SPListItem object, 202
- SPQuery object, 221

Folder Scope function, 252

Folder views, 40

Folders. *See also* SPFolder object.

- creating, 467
- discussion topics, 467
- object models, 155
- Standard View, 28
- summarizing tasks, 467

Folders, list items

- adding, 207
- counting, 207
- deleting, 207, 208

- ensuring currency, 208
 - getting as DataTable objects, 207
 - getting by collection position, 207
 - getting by integer ID, 207
 - reordering, 208
 - Folders, lists. *See also* SPFolder object.
 - converting to XMLDATA format, 207
 - counting, 196
 - fields, returning, 207
 - folder objects, 155
 - parent collection, returning, 207
 - returning, 195
 - Folders property
 - SPList object, 195
 - SPWeb object, 188
 - Folders setting, 40
 - Fonts, site settings, 70
 - Foreign Identifier Association Entity property, 393
 - Foreign Identifier Association Name property, 393
 - Foreign Identifier property, 370
 - Foreign key association, 443
 - Foreign key identifier, 370
 - Foreign key-based association, 430
 - Foreign key-less association, 430
 - Form action buttons, 254
 - Form Action function, 254
 - Form content type, 466
 - FormatDate() method, 229
 - Formatting
 - date and time, 229
 - Standard View, 16, 19, 20
 - Forms
 - entering data into, 466
 - libraries of, 34
 - workflows, 552–562
 - Forms Based Authentication, enabling, 78, 79
 - Forms property, 195
 - Formula function, 251
 - Formulas, inserting, 251
 - Full Control permission, 28–29, 693
 - Full deployment, 452–454
 - Full Path property, Elements.xml file, 130
 - FullUrl property, 328
- G**
- Galleries
 - displaying, 98, 101
 - site settings, 67–68
 - Garbage collection, 157. *See also* Disposing of managed objects.
 - Gates, Bill, 2–3
 - General Settings section, 24–28
 - Generic List type, 364
 - GetDataTable() method
 - SPList object, 197
 - SPListItemCollection object, 207
 - GetGenericSetupPath() method, 228
 - GetItemId() method
 - SPList object, 195
 - SPListItemCollection object, 207
 - GetItems() method, 195
 - GetLimitedWebPartManager() method, 214
 - GetLocalizedString() method, 229
 - GetSiteData() method, 189
 - Global assembly cache, deploying projects to, 118
 - Global Resources File project items, 125
 - Granting permission levels, 63
 - Group attribute, 473
 - Group By setting, 40
 - Group property, 217
 - Group work
 - circulation, creating, 467
 - holidays, adding, 467
 - lists, adding new words, 467
 - new notifications, adding, 468
 - official notices, adding, 468
 - phone call memos, adding, 468
 - resource groups, adding, 468
 - resources, adding, 468
 - timecard data, adding, 468
 - users, adding, 468
 - Group Work content types, 467–468
 - Group Work Site subsites, 54
 - Grouping
 - items in custom list views, 251
 - workflow activities, 529
 - Groups
 - management, users and permissions, 60–61
 - membership permissions, application pages, 694–696
 - names, specifying, 473
 - permission level settings, 60–62
 - Groups property, 226
 - GUID, changing, 345
- H**
- Handle ItemAdded property, 138
 - Handle ItemAdding property, 138
 - Handle ItemDeleted property, 138
 - Handle ItemDeleting property, 138
 - HandleAccessDenied() method, 229
 - HandleExternalEvent workflow activity, 530
 - Headings for visual web parts, size and color, 586–587

Health Analyzer command group, 90
 Height function, 255
 Help Settings, 74
 Hidden property
 SPField object, 217
 SPList object, 195
 Hide Custom Action, SharePoint element,
 707, 709
 Hide from browser setting, 245
 Hide function, 255
 Hiding web parts, 255
 Hive, deploying projects to, 118–121
 Holiday content type, 467
 Holidays, adding, 467
 Host sites
 creating, 169
 external, getting/sending data, 530
 specifying, 78
 web part hosts, 466
 HS Web Site setting, 78
 HTML Form web parts, 50
 HTTP Request Monitoring and Throttling,
 81
 HTTP requests, 228–229

I

ID attribute, 473
 ID for content types, specifying, 473, 482
 ID property
 SPList object, 194
 SPListItem object, 201
 SPUser object, 226
 SPWeb object, 187
 Id property, 217
 Identifier Entity property, 393
 Identifier property
 Data Elements, 370
 type descriptors, Visual Studio external
 content types, 393
 Identifiers
 listing, 370
 mapping, 441–442
 renaming, 408
 IDisposable objects, 157–160
 IDs, features, 700
 If/Else constructs, 530
 IfElse workflow activity, 530
 If-Then-Else expressions, storing, 532
 Image Alt Text property, 727
 Image attribute, 275
 Image Url property, 727
 Image Viewer web parts, 50–51
 IMAGES folder, deploying projects to, 120
 Import Reusable Workflow projects, 109

Import SharePoint Solution Package
 projects, 109
 Importing
 BDC models from SharePoint Designer,
 385
 list instances. *See* List instances, importing.
 spreadsheet lists, 15
 workflows to Visual Studio, 570–578
 .WSP file projects, 109
 Incremental deployment, 452–454
 IndependentValuePath property, 627
 InDocumentLibrary property, 214
 Information Rights Management, enabling,
 93
 Inheritance
 by copy, list instances, 259
 permission levels, 62
 Visual Studio deployment configurations,
 771
 Inherits attribute, 473
 Initialize() method, 761–765
 InitializeWorkflow workflow activity, 533
 Initializing workflows, 533
 Initiation data, list workflow events, 329
 InitiationData property, 329
 Inline editing, enabling, 40, 252
 Inline Editing function, 252
 Inline Editing setting, 40
 Insert Control function, 254
 Installing
 features, 710–711
 SharePoint Designer, 807–808
 Visual Studio, 803–807
 Visual Studio 2010 SDK, 760
 Visual Studio custom deployment,
 766–770
 Installing SharePoint
 basic steps, 789
 on a client operating system, 789–794
 configuration, 796–803
 operating system requirements, 787–788
 on a server operating system, 794–795
 InstanceId property, 329
 InternalName property, 217
 InvalidateListItem() method, 325
 InvalidateWeb() method, 325
 Invalidating a list item or web, 325
 InvokeWebService workflow activity, 532
 InvokeWorkflow workflow activity, 532
 Is Cached property, 394, 396
 Is Collection property, 394
 Is Hidden property, 727
 ISO8061 date and time format, 228
 IsSelectionEnabled property, 627

- Issue content type, 469
 - Issue Tracking, list template, 234
 - Issue tracking lists, 15
 - Item content type, 469
 - Item limit, document views, 40
 - Item Limit setting, 40
 - Item property
 - SPContext object, 173
 - SPFile object, 213
 - SPFolder object, 210
 - SPListItem object, 201
 - ItemAdding event handler, 161–162, 166
 - ItemBinding property, 627
 - ItemCount property
 - SPFolder object, 211
 - SPList object, 196
 - ItemDeleting event handler, 168
 - ItemId property, 173
 - Item() method, 207
 - Item-related workflow activities, 532
 - Items, creating, 469
 - Items property, 194
 - Iterating through elements, 531
- J**
- Join() method, 177
- L**
- Label1_Load handler, 164
 - Language locale, returning, 188
 - Language property, 188
 - LAYOUTS folder, deploying projects to, 120
 - Length property, 213
 - Level, returning
 - files, 213
 - list items, 202
 - Level property
 - SPFile object, 213
 - SPListItem object, 202
 - Libraries
 - definition, 33
 - folder objects, 155
 - forms, 34
 - pictures, 34
 - types of, 34
 - Wiki pages, 34
 - Libraries, document
 - checking documents in and out, 37
 - columns, displaying and arranging, 37, 40
 - content types, managing, 43–46
 - copying documents, 37
 - creating, 34–36, 497
 - definition, 34
 - Documents ribbon tab, 37
 - downloading documents, 37
 - Edit View page, 39
 - editing documents, 37
 - Library Settings page, 42–46
 - revision history, 37
 - versioning, 35
 - view settings, 39–40
 - views, 37–41
 - Library Settings page, 42–46
 - Limit filter, 374
 - Limited Access permission, application
 - pages, 693
 - Link content type, 469
 - Link lists, 15
 - Link to a Document content type, 466
 - Links
 - creating, 469
 - between documents, 466
 - LINQ to SharePoint (SPLINQ), querying
 - lists, 307–315
 - LINQ to SQL, 403
 - Lippert, Eric, 520
 - List associations, workflows, 548–552
 - List columns. *See also* SPField object.
 - adding, 242–246
 - custom, *vs.* site column definitions, 243
 - default value, getting and setting, 218
 - definition, 232, 459
 - deleting, 217
 - deriving other fields from, 218
 - description, returning, 217
 - display name, returning, 218
 - display size, returning, 218
 - edit form, displaying, 218
 - vs.* fields, 235
 - group of site columns, returning, 217
 - GUID, returning, 217
 - internal name, getting and setting, 218
 - internal name, returning, 217
 - name, returning, 217
 - parent list, returning, 218
 - rendering, 218
 - required, setting, 217
 - schema XML, getting and setting, 218
 - scope, returning, 218
 - sealing, 218
 - site-relative path, returning, 218
 - SPMobileBaseFieldControl class, returning, 218
 - title, returning, 217
 - type, returning, 217
 - updating, 217
 - version, returning, 217
 - visibility, setting, 217

- List columns, site columns
 - vs.* custom list columns, 235
 - definition, 234, 459
 - vs.* fields, 235
- list command, 2–4
- List content types, 468–469. *See also specific content types.*
- List Definition From Content Type project items, 123
- List Definition project items, 123
- List Definition projects, 108, 109
- List Definitions, 778
- List definitions
 - creating from content types, 487–491
 - current project *vs.* local debugging site, 282
 - definition, 232
 - vs.* list templates, 232
- List definitions, creating with Visual Studio.
 - See also* Elements.xml file.
 - default schema, defining, 275
 - description, specifying, 275, 279
 - display name, specifying, 275
 - instance title, specifying, 279
 - instance URL, specifying, 279
 - instances, displaying on Quick Launch bar, 275, 279
 - item level permissions, 275
 - list icon URL, specifying, 275
 - local debugging site, specifying, 272
 - name, specifying, 273
 - overview, 272
 - project structure, displaying, 274
 - for sandboxed solutions, 273
 - Schema.xml file, modifying, 276–278
 - template name, specifying, 275
 - template type, specifying, 279
 - templates, ordering priority, 275
 - type, specifying, 273
 - unique identifier, specifying, 275
- List e-mail event receivers, 126
- List event receivers, 126
- List events
 - feature GUID, 321
 - FeatureId property, 321
 - Field property, 321
 - FieldName property, 321
 - FieldXml property, 322
 - list of, 321
 - List property, 322
 - ListId property, 322
 - ListTitle property, 322
 - SPListEventProperties object, 321–322
 - SPListEventReceiver, 321–322
 - TemplateId property, 322
 - user triggering, 322
 - UserDisplayName property, 322
 - UserLoginName property, 322
 - web GUID, 322
 - web URL, 322
 - WebId property, 322
 - WebUrl property, 322
- List events, affected fields
 - name, 321
 - SPField instance, 321
 - XML definition, 322
- List events, affected lists
 - GUID, 322
 - list template ID, 322
 - SPList instance, 322
 - title, 322
- List Instance, 707, 709, 776
- List Instance project items, 123
- List Instances, 778
- List instances
 - adding to existing projects, 280–284
 - creating with Visual Studio. *See* List definitions, creating with Visual Studio.
 - definition, 232–233
 - deploying, 283
 - inheritance by copy, 259
- List instances, creating with SharePoint Designer
 - columns, adding, 242–246
 - customizing the instance, 241–242
 - overview, 236–238
 - site columns, adding, 242–244
 - from a template, 238–241
- List instances, importing dependencies, 270–271
 - to an existing project, 262
 - to a new project, 262–267
 - to Visual Studio, 261–271
 - Windows 7 workaround, 264
 - Windows Vista workaround, 264
- List Item Event handler, adding to farm solutions, 163–166
- List Item event receiver, adding, 167–168
- List item events
 - absolute web URL, 325
 - after properties, 324
 - InvalidateListItem() method, 325
 - InvalidateWeb() method, 325
 - invalidating a list item or web, 325
 - item context, 324
 - item SPList instance, 324
 - item URL, after event, 324
 - item URL, before event, 324

- list instance ID, 324
- list item ID, 325
- list of, 323
- list title, 325
- opening a site and web, 325
- OpenSite() method, 325
- OpenWeb() method, 325
- overview, 322
 - before properties, 324
 - property name, 324
 - server-relative web URL, 325
 - SPIItemEventReceiver, 322–327
 - SPList item instance, 325
 - SPUrlZone enumeration, 325
 - triggering user, display name, 325
 - triggering user, ID, 324
 - triggering user, login name, 325
 - version tracking, 325
- List item events, SPIItemEventProperties object
 - AfterProperties property, 324, 325–326
 - AfterUrl property, 324
 - BeforeProperties property, 324, 325–326
 - BeforeUrl property, 324
 - Context property, 324
 - CurrentUserId property, 324
 - FileSystemObjectTypePropertyName property, 324
 - List property, 324
 - ListId property, 324
 - ListItem property, 325
 - ListItemId property, 325
 - ListTitle property, 325
 - RelativeWebUrl property, 325
 - UserDisplayName property, 325
 - UserLoginName property, 325
 - Versionless property, 325
 - WebUrl property, 325
 - Zone property, 325
- List items. *See also* SPListItem object; SPListItemCollection object.
 - adding, 195, 207
 - attachments, returning, 202
 - auditing, 203
 - content type, returning, 203
 - converting to XMLDATA format, 201, 207
 - copying, 202, 203
 - counting, 196, 207
 - creating, 31–33
 - DataTable objects, returning, 207
 - default opening method, setting, 196
 - deleting, 202, 207–208
 - display name, returning, 203
 - editing, 31–33
 - ensuring currency, 208
 - event receivers, 126
 - field values, getting and setting, 201
 - fields, returning, 202, 207
 - file object, returning, 202
 - folder object, returning, 202
 - getting, 207
 - GUID, returning, 203
 - ID, returning, 201
 - level, returning, 202
 - names, getting and setting, 201
 - object models, 154
 - parent list, returning, 203, 207
 - parent list collection, returning, 203
 - properties, returning, 202
 - recycling, 203
 - reordering, 208
 - role inheritance, setting, 203
 - site association, returning, 201
 - site-relative URL, returning, 201
 - SPListItem objects, returning, 194
 - titles, returning, 202
 - updating, 202, 203
 - user arrangement, enabling, 196
 - version history, returning, 202
 - viewing, 31–32
 - workflow tasks, returning, 202
- List property
 - SPContext object, 172
 - SPIItemEventProperties object, 324
 - SPListEventProperties object, 322
 - SPListItemCollection object, 207
- List Settings page
 - Advanced Settings page, 27–28
 - General Settings section, 24–28
 - Permissions and Management section, 28–30
 - Versioning settings, 25–26
- List tab, 22
- List Template, SharePoint element, 707, 709
- List templates
 - Announcements, 233
 - Calendar, 233
 - Contacts, 233
 - custom, 232
 - Custom List, 234
 - Custom List in Datasheet View, 234
 - definition, 232
 - Discussion Board, 234
 - Document Library, 234
 - Issue Tracking, 234
 - Links, 234
 - vs.* list definitions, 232
 - missing dependencies, 261

- List templates, *continued*
 - Project Tasks, 234
 - Survey, 234
 - Tasks, 234
 - user-created, 232
- List Templates setting, 68
- List Unique Permissions Threshold setting, 81
- List View Style content type, 466
- List view styles, creating, 466
- List View Threshold setting, 81
- List View Tools
 - Add Connection function, 252, 255
 - Add/Remove Columns function, 252
 - Asynchronous Update function, 252
 - Chrome State function, 255
 - Chrome Type function, 254
 - Close function, 255
 - Conditional Formatting function, 251
 - Connections function, 255
 - Customize XSLT function, 254
 - Data Source Details function, 252
 - Data View Preview function, 254
 - Design tab functions, 253–254
 - Edit in personal view function, 255
 - To File function, 255
 - Filter function, 251
 - Finder function, 251
 - Folder Scope function, 252
 - Form Action function, 254
 - Formula function, 251
 - Height function, 255
 - Hide function, 255
 - Inline Editing function, 252
 - Insert Control function, 254
 - Manage Connection function, 255
 - Manage Connections function, 252
 - Minimize function, 255
 - Options function, 253
 - Options tab functions, 251–253
 - overview, 249
 - Paging function, 251
 - Parameters function, 251
 - Properties function, 254
 - Refresh Button function, 252
 - Refresh Data function, 252
 - Refresh Interval function, 252
 - Revert XSLT function, 254
 - Sample Data function, 253
 - To Site Gallery function, 255
 - Sort & Group function, 251
 - Table tab, 252–253
 - Totals Row function, 253
 - View Style function, 253
 - Web Part tab functions, 254–255
 - Web Part Title function, 254
 - Width function, 255
 - Zone Change function, 255
- List views
 - columns, displaying and arranging, 19, 21
 - Datasheet View, 16, 18
 - default, 16, 248
 - definition, 235
 - formats, list of, 19, 20
 - spreadsheet-like, 16
 - table-based, 253
 - web-based, 16
- List views, custom
 - creating. *See also* List View Tools.
 - asynchronous updates, enabling, 252
 - check boxes, inserting, 254
 - columns, adding/removing, 252
 - conditional formatting, 251
 - connection properties, changing, 252
 - connections, adding, 252
 - data source details, displaying, 252
 - data view controls, inserting, 254
 - default finder method, 251
 - default page, 248
 - filtering, 251
 - form action buttons, 254
 - formulas, inserting, 251
 - grouping items, 251
 - inline editing, enabling, 252
 - overview, 249
 - paging criteria, 251
 - previewing data, 254
 - query string parameters, defining, 251
 - Refresh button, adding, 252
 - refresh interval, setting, 252
 - refreshing data display, 252
 - sample data, generating, 253
 - sorting items, 251
 - text boxes, inserting, 254
 - toolbar options, 253
 - totals row, displaying, 253
 - view scope, specifying, 252
 - view style, specifying, 253
 - XSLT, customizing, 254
 - XSLT, reverting to default, 254
 - deploying, 255–261
 - displaying with web parts
 - chrome type, setting, 254
 - chrome type, state of, 255
 - closing web parts, 255
 - connecting web parts, 255
 - editing web parts, 255
 - hiding web parts, 255

- minimizing web parts, 255
- properties dialog, 254
- saving web parts to files, 255
- saving web parts to site galleries, 255
- title, setting, 254
- web part connections, adding, 255
- web part connections, properties, 255
- web part height, setting, 255
- web part width, setting, 255
- web part zone, changing, 255
- saving as a template, 255–261
- List views, Standard View
 - creating. *See also* List Settings page.
 - attachments, 27
 - choosing a format, 16, 19, 20
 - content types setting, 27
 - datasheet view editing, enabling, 28
 - dialogs, enabling, 28
 - folders, 28
 - item-level permissions, 27
 - naming the view, 19
 - offline client availability, 28
 - permissions, 28–30
 - search results, customizing, 28
 - creating columns
 - arranging, 19
 - Create Column dialog, 23, 24
 - custom site columns, 27
 - List tab, 22
 - reusing column definitions, 22, 24, 26
 - from scratch, 22
 - selecting, 19
 - settings, illustration, 21
 - description, 16
 - illustration, 17
 - reverting to a previous version, 25
 - tracking changes, 25–26
- List workflow events
 - absolute site URL, 330
 - activation properties, 329
 - ActivationProperties property, 329
 - association data, 329
 - AssociationData property, 329
 - completion type, 329
 - CompletionType property, 329
 - ErrorException property, 329
 - exception information, 329
 - initiation data, 329
 - InitiationData property, 329
 - instance GUID, 329
 - InstanceId property, 329
 - list of, 328–329
 - overview, 328–329
 - postponed events, 329
 - PostponedEvent property, 329
 - RelativeWebUrl property, 329
 - server-relative site URL, 329
 - SPWorkflowEventProperties object, 329–330
 - SPWorkflowEventReceiver, 328–330
 - TerminatedByUserId property, 330
 - terminating user, ID, 330
 - WebUrl property, 330
- Listen workflow activity, 530
- Listening to external events, 530
- ListId property
 - SPContext object, 173
 - SPItemEventProperties object, 324
 - SPListEventProperties object, 322
- ListItem property
 - SPContext object, 173
 - SPItemEventProperties object, 325
- ListItemCollectionPosition property, 207
- ListItemId property, 325
- ListItems property, 203
- Lists. *See also* SPList object; *specific lists*.
 - adding new words, 467
 - associating with workflows, 536–540
 - attachments, allowing, 245
 - auditing, 195
 - author, returning, 195
 - base template type, returning, 196
 - base type, returning, 196
 - blog posts, creating, 469
 - calendar, description, 14
 - calendars, template for, 233
 - comments, creating, 468
 - contact information, storing, 468
 - contacts, description, 14
 - contacts, template for, 233
 - content approval, requiring, 246
 - content database, updating, 194
 - contents, modifying, 154
 - creating, 15
 - creation timestamp, returning, 194
 - custom in datasheet view, description, 15
 - custom in datasheet view, template for, 234
 - deadlines, creating, 469
 - default view, returning, 196
 - deleting, 28, 195
 - description, getting and setting, 194
 - discussion board, description, 15
 - discussion board, template for, 234
 - displaying, 98, 101
 - displaying in the Quick Launch menu, 245
 - document libraries, template for, 234
 - event receivers, returning, 196

Lists, *continued*

- events, creating, 469
 - external, 15
 - external data source, returning, 196
 - fields, returning, 194
 - forms, returning, 195
 - GUID, returning, 173, 194
 - hiding from the browser, 245
 - import spreadsheet, 15
 - items, creating, 469
 - meetings, creating, 469
 - messages, creating, 469
 - overview, 12, 14–15
 - parent list collection, returning, 194
 - parent SPWeb object, returning, 195
 - permission level, 65
 - permissions, specifying maximum number of, 81
 - project tasks, description, 15
 - project tasks, template for, 234
 - putting in the Recycle bin, 195
 - query results, returning, 197
 - Quick Launch access, getting and setting, 196
 - relationships, 315
 - reserving resources, 469
 - role assignments, returning, 195
 - role inheritance, setting, 196
 - root folder, returning, 195
 - saving as templates, 28
 - scheduling appointments, 469
 - settings, 245–246
 - survey, description, 15
 - survey, template for, 234
 - template types, 489–490
 - titles, getting and setting, 194
 - tracking issues, 469
 - types of, 15
 - URL for default view, returning, 197
 - user-created folders, allowing, 245
 - version number, returning, 194
 - versioning, 246
 - views, 16
 - views, returning, 195
 - visibility, getting and setting, 195
 - work items, creating, 469
 - workflow associations, returning, 197
- Lists, announcements
- creating, 468
 - description, 14
 - template for, 233
- Lists, content types
- Content Types setting, 27
 - definition, 235
 - enabling, 246–248
 - multiple, 246
 - returning, 196
- Lists, creating programmatically
- application pages, 285–294
 - ASPX files, 288
 - client-side object model, 284, 294–300
 - .NET 3.5 Framework Client profile, 285
 - object model naming, 290
 - overview, 284
 - sandboxed solutions, 286
 - server-side object model, 284, 285–294, 298
- Lists, custom
- creating, 16
 - description, 14
 - template for, 234
- Lists, issue tracking
- description, 15
 - template for, 234
- Lists, links
- creating, 469
 - description, 15
 - template for, 234
- Lists, modifying programmatically
- client-side object model, 305–307
 - overview, 300
 - server-side object model, 300–304
- Lists, querying
- Add Service Reference feature, 309–311
 - with CAML, 307–308
 - entity classes, generating, 308–311
 - with SPLINQ (LINQ to SharePoint), 307–315
 - SPMetal.exe tool, 308–309
 - strongly-typed queries, 307–315
- Lists, tasks
- creating, 469
 - description, 15
 - template for, 234
- Lists and Libraries subnode, 98, 101
- Lists property
- SPList object, 194
 - SPSiteDataQuery object, 224
 - SPWeb object, 187
- ListTemplates property, 189
- ListTemplates subnode, 98, 100
- ListTitle property
- SPItemEventProperties object, 325
 - SPListEventProperties object, 322
- LOB Name property, 394
- LobSystemInstance, indexing, 451
- Local debugging site, specifying, 272
- Local property, 177

- Locale property, 189
- Localized strings, returning, 229
- Locking site collections, 85
- Logging history list, 533
- Login name, returning, 226
- LoginName property, 226
- Logs, setting expiration period, 82
- LogToHistoryListActivity workflow activity, 533
- Look and feel. *See* Site settings, look and feel.
- Lookup() method, 179
- M**
- Mail Settings, 82
- Manage Alerts permission level, 67
- Manage Antivirus Settings, 92
- Manage Connection function, 255
- Manage Connections function, 252
- Manage Features command, 82
- Manage Lists permission level, 65
- Manage permissions permission level, 66
- Manage Personal Views permission level, 67
- Manage Site Features action, 72
- Manage the Farm Administrators Group command, 92
- Manage Web Part Security, 93
- Manage Web Site permission level, 66
- Managed accounts, configuring, 92
- Managed Path command, 82
- Manual feature activation, 711
- Map to Identifier property, 370
- MapFile, XML element, 705
- Mapped folders
 - associating documents with content type, 507
 - deploying projects, 117–121
 - Visual Studio feature designer, 733–739
- Master Page content type, 466
- Master Page Setting for Application_Layout Pages, 81
- Master pages
 - architecture, 658–659
 - ASP.NET master pages, adding, 669
 - creating, 661–671
 - customizing, 661–671
 - default, setting, 667
 - deploying, 671
 - primary, 661
 - v4.master page, 661, 664
- Master Pages setting, 68
- MasterUrl property, 189
- Maximum Upload Size setting, 81
- Meetings, creating, 469
- Memory management. *See* Disposing of managed objects; Garbage collection.
- Message content type, 469
- Messages, creating, 469
- MetaWeblog API, enabling, 80
- Method instance, Visual Studio external content types
 - custom properties, 396
 - default, 396
 - name, 396
 - return parameter name, 396
 - return type descriptor, 396
 - type, 396
- Method instance properties, Visual Studio external content types, 396. *See also specific properties.*
- Method instance types, Visual Studio external content types, 395. *See also specific types.*
- Method signatures, 395
- Methods, Visual Studio external content types
 - creating, 389
 - default, renaming, 408–409
 - deleting, 389
 - displaying method definitions, 389
 - implementing, 426–428
 - method parameter directions, 389–390
- Minimize function, 255
- Minimizing web parts, 255
- Misc property category
 - Elements.xml file, 129–130
 - EventReceivers folder, 137
- Mobile device views, 40
- Mobile setting, 40
- Modification events, workflows, 534
- Module, SharePoint element
 - conflict resolution, 776
 - description, 707
 - sandboxed solutions, 778
 - scope, 709
- Module project items, 123–124
- Module projects, 108, 109
- Modules property, 189
- Monitoring
 - HTTP requests, 81
 - server health, 88–91
- MoveTo() method
 - SPFile object, 214
 - SPFolder object, 211
- Multi-list workflows, 552
- Multipage Meeting Workspace subsites, 54, 56–57

N

- Name attribute, 275, 473
 - Name property
 - method instance, Visual Studio external content types, 396
 - SPFile object, 213
 - SPFolder object, 210
 - SPListItem object, 201
 - SPUser object, 226
 - SPWeb object, 187
 - type descriptors, Visual Studio external content types, 394
 - Visual Studio projects, package designer, 746
 - Name subelement, 132
 - Names
 - displaying, visual web parts, 603
 - features, 700
 - Names, getting and setting
 - list items, 201
 - users, 226
 - Names, returning
 - fields, lists, 217
 - files, 213
 - web applications, 180
 - Names, specifying
 - content types, Visual Studio, 473
 - list definitions, 273
 - Naming Standard Views, 19
 - NativeErrorMessage property, 228
 - NativeStackTrace property, 228
 - Navigation, sites, 188
 - Navigation property, 188
 - .NET 3.5 Framework Client profile, 285
 - .NET code, executing, 530
 - .NET external source type, 366
 - .NET Type, 370
 - .NET Type property, 370
 - New Word content type, 467
 - NewServerRelative Url property, 328
 - no activation deployment, 751, 753–754
 - NoDeployment deployment type, 723, 724
 - Notes, getting and setting, 226
 - Notes property, 226
 - Notifications, adding, 468
 - Numeric field settings, regional, 189
- O**
- Object model naming, 290
 - Object models, Server sandboxed solutions, 160–162
 - Object models, SharePoint
 - context information, 154
 - disposing of objects, 155
 - folders, 155
 - key objects, 154–156. *See also specific objects.*
 - list contents, modifying, 154
 - list items, 154
 - object relationship diagram, 156
 - objects, disposing of, 155
 - overview, 153–154
 - site collections, 154
 - sites, 154
 - users, 154
 - utility class, 156
 - Objects, disposing of, 155
 - Office Item type associations, 370
 - Office Item Types, 364
 - Office Property property, 370
 - Official Notice content type, 468
 - Official notices, adding, 468
 - Offline client availability, 28
 - Online presence information, enabling, 80
 - OnQuickLaunch attribute, 275, 279
 - OnQuickLaunch property, 196
 - OnTaskChanged workflow activity, 533
 - OnTaskCreated workflow activity, 533
 - OnTaskDeleted workflow activity, 533
 - OnWorkflowActivated workflow activity, 534
 - OnWorkflowItemChanged workflow activity, 534
 - OnWorkflowItemDeleted workflow activity, 534
 - OnWorkflowModified workflow activity, 534
 - Open Items permission level, 65, 67
 - OpenBinary() method, 213
 - Open() method, 177
 - OpenSite() method, 325
 - OpenWeb() method, 325
 - Options function, 253
 - Options tab functions, 251–253
 - Ordered property, 196
 - Override Checkout permission level, 65
- P**
- Package designer. *See* Visual Studio projects, package designer.
 - Package information, Visual Studio custom deployment, 765–766
 - PackageOnly deployment type, 723
 - Packages folder, workflows, 541–542
 - Packaging Explorer, 747–748
 - Packaging solutions. *See* Solution packaging.
 - @Page directive, 679
 - Page directive, application pages, 686

- Page Number filter, 374
- Page type pages, 46–47
- Page Viewer web parts, 50
- Pages. *See also* SharePoint pages.
 - page type, 46–47
 - types of, 46–47
 - web parts. *See* Web Part Page type; Web parts.
 - wiki, 46–47
- Paging criteria, custom list views, 251
- Paging function, 251
- Parallel workflow activity, 531
- Parameter, XML element, 705
- Parameters function, 251
- ParentFolder property, 211
- ParentList property
 - SPField object, 218
 - SPListItem object, 203
- ParentListID property, 211
- ParentWeb property
 - SPFolder object, 211
 - SPList object, 195
 - SPWeb object, 188
- ParentWebId property, 328
- ParseDate() method, 229
- Partial updating of visual web parts, enabling, 586, 597–600
- Passwords
 - automatic changing, 92
 - change settings, 92
 - including in e-mail, 81
- Path management, specifying, 82
- Permission levels, validating, 692–694
- Permission policies, setting, 83
- Permission Policy command, 83
- Permissions. *See also* Security; Site settings, users and permissions; SPSecurity object.
 - application pages, 692–696
 - AspNetHostingPermissions, 653
 - deploying projects, 110
 - external content, 378–380
 - for external content, 378–380
 - item level, list definitions, 275
 - item-level, 27
 - lists, specifying maximum number of, 81
 - SecurityPermissions, 653
 - SharePointSecurityPermission, 653
 - Standard View, 28–30
 - SystemSecurityPermission, 653
 - web part configuration, 653
- Permissions, lists
 - Contribute level, 28–29
 - custom groups, 29
 - custom permission sets, 29
 - Full Control level, 28–29
 - group members, adding/removing, 29–30
 - group types, 28–29
 - levels, 28–29
 - Read level, 28–29
 - Team Site Members, 28–29
 - Team Site Owners, 28–29
 - Team Site Visitors, 28–29
- Permissions and Management section, 28–30
- Personal permissions, 66
- Personalizable attribute, 603
- Phone Call Memo content type, 468
- Phone call memos, adding, 468
- Picture content type, 466
- Picture libraries, 34, 466, 476–480
- Picture Library Slideshow web parts, 50
- Pie chart, creating, 620–630
- PieSeries control, 623–627
- PlaceholderAdditionalPageHead placeholder, 687
- PlaceholderMain placeholder, 687
- PlaceholderPageTitle placeholder, 687
- PlaceholderPageTitleInTitleArea placeholder, 687
- Playing back site changes, 5
- Policies property, 180
- Policy workflow activity, 532
- Portal Site Connection command, 74
- Post content type, 469
- Post (asynchronous) events, 319
- Post list type, 364
- Postponed events, 329
- PostponedEvent property, 329
- Prepopulated project types. *See* Project types, prepopulated.
- Presence Settings, 80
- Pre-Updater Field property, 394
- Previewing data, custom list views, 254
- Problems, reviewing, 90
- Products property, 177
- Project item types
 - Application Page, 122
 - Business Data Connectivity Model, 122
 - Business Data Connectivity Resource Item, 125
 - Content Type, 122–123
 - Empty Element, 123
 - Event Receiver, 123
 - Global Resources File, 125
 - List Definition, 123
 - List Definition From Content Type, 123
 - List Instance, 123

- Project item types, *continued*
 - Module, 123–124
 - Sequential Workflow, 124
 - State Machine Workflow, 124
 - User Control, 124
 - Visual Web Part, 124
 - Web Part, 124
 - Workflow Association Form, 125
 - Workflow Initiation Form, 125
- Project items
 - adding, 121–127, 730
 - creating, 126–127
 - Element Manifest file. *See* Elements.xml file.
 - overview, 128
 - root item node or folder. *See* EventReceiver1 folder.
- Project items, code file (EventReceiver1.cs)
 - code sample, 133–134, 134–135
 - overview, 133
- Project Output References, 138
- Project task lists, 15
- Project Tasks, list template, 234
- Project types, empty
 - debug assemblies, 114
 - description, 107
 - Empty SharePoint Project, 107
 - release assemblies, 114
 - startup settings, 114
 - structure, 113
- Project types, prepopulated
 - Business Data Connectivity Model, 108, 109
 - Content Type, 108, 109
 - Event Receiver, 108, 109
 - Import Reusable Workflow, 109
 - Import SharePoint Solution Package, 109
 - importing a .WSP file, 109
 - List Definition, 108, 109
 - Module, 108, 109
 - Sequential Workflow, 108, 109
 - Site Definition, 109, 109
 - State Machine Workflow, 108, 109
 - Visual Web Part, 107, 109
- Projects. *See also* Solutions.
 - AllowPartiallyTrustedCallers attribute, 116
 - attributes, 116
 - building, 144–148
 - debugging, 148–152
 - Elements.xml file, 147–148
 - features, 139–143
 - Features folder, 116
 - Feature.xml file, 147
 - file name, 115
 - folder name, 115
 - Manifest.xml file, 146–147
 - mapped folders, 117–121
 - multiple, 110
 - Package Designer, 143–144
 - package files (.WSP), creating, 116
 - Package folder, 116
 - packaging, 115
 - referenced assemblies, 116
 - sandboxed *vs.* farm, 111–113
 - single, 109–110
 - startup item, 115
 - structuring, 109–110
 - trust level, specifying, 110–111
 - Visual Studio. *See* Visual Studio projects.
- Projects, deployment
 - Active Deployment Configuration property, 115
 - Assembly Deployment Target property, 115
 - to the global assembly cache, 118
 - to the hive, 118–121
 - to the IMAGES folder, 120
 - to the LAYOUTS folder, 120
 - mapped folders, 117–121
 - permissions, 110
 - to the SiteTemplates folder, 120
 - to specific web application directories, 118
 - target URL, specifying, 115
 - to the TEMPLATE folder, 120
- Projects, properties
 - Active Deployment Configuration, 115
 - Assembly Deployment Target, 115
 - Include Assembly in Package, 115
 - Project File, 115
 - Project Folder, 115
 - Sandboxed Solution, 115
 - Site URL, 115
 - Startup Item, 115
- Properties
 - Active On Default, 727
 - Always Force Install, 727
 - Auto Activate In Central Admin, 727
 - Creator, 727
 - Default Resource File, 727
 - defining, features, 704
 - Deployment Path, 727
 - Description, 727
 - event receivers. *See* Event receiver properties.
 - Feature Id, 727
 - Image Alt Text, 727

- Image Url, 727
 - Is Hidden, 727
 - Receiver Assembly, 727
 - Receiver Class, 727
 - Requires Resources, 727
 - Scope, 727
 - Solution Id, 727
 - Title, 728
 - type descriptor. *See* Type descriptor properties.
 - UIVersion, 728
 - Upgrade Actions Receiver Assembly, 728
 - Upgrade Actions Receiver Class, 728
 - Version, 728
 - Properties, Elements.xml file
 - Advanced, 129–130
 - BuildAction, 130
 - Copy to Output Directory, 130
 - Custom Tool, 130
 - Custom Tool Namespace, 130
 - Deployment Location, 130
 - Deployment Type, 130
 - File Name, 130
 - Full Path, 130
 - Misc, 129–130
 - SharePoint, 129–130
 - Properties, EventReceiver1 folder
 - Assembly, 138
 - Class Name, 138
 - Feature Properties, 137
 - Feature Receiver, 138
 - Folder Name, 137
 - Handle ItemAdded, 138
 - Handle ItemAdding, 138
 - Handle ItemDeleted, 138
 - Handle ItemDeleting, 138
 - Misc, 137
 - SharePoint, 137
 - SharePoint Events, 138–139
 - Properties, projects
 - Active Deployment Configuration, 115
 - Assembly Deployment Target, 115
 - Include Assembly in Package, 115
 - Project File, 115
 - Project Folder, 115
 - Sandboxed Solution, 115
 - Site URL, 115
 - Startup Item, 115
 - Properties, returning
 - files, 213
 - list items, 202
 - Properties dialog, 254
 - Properties function, 254
 - Properties property
 - SPFolder object, 210
 - SPListItem object, 202
 - SPWeb object, 187
 - Property, XML element, 704
 - Providers, web part connections, 637
 - Provision() method, 180
 - Provisioned property, 189
 - Provisioning status, returning, 189
 - Provisioning web applications, 180
 - Public URL setting, 78
 - Publishing method, returning, 213
 - Publishing workflows, 569
 - Publish() method, 213
- Q**
- Queries. *See also* SPListItemCollection object;
SPQuery object; SPSiteDataQuery object.
 - folder restrictions, 221
 - getting and setting in CAML, 221, 224
 - large, specifying a window for, 81
 - lists. *See* Lists, querying.
 - restricting, 221, 224
 - returned fields, getting and setting, 221, 224
 - row limits, setting, 221, 224
 - sub webs, including, 224
 - view attributes, getting and setting, 221
 - Query property
 - SPQuery object, 221
 - SPSiteDataQuery object, 224
 - Query string parameters, defining, 251
 - Quick Launch access, getting and setting, 196
 - Quick launch area, accessing, 188
 - Quick Launch setting, 70–71
 - Quota templates, modifying, 85
 - Quotas for default site collections, 80
- R**
- Read permission, 28–29, 693
 - Read-only, Data Source Elements, 370
 - Read-Only property
 - Data Elements, 370
 - type descriptors, Visual Studio external content types, 394
 - ReceiveActivity workflow activity, 532
 - Receiver Assembly property, 345, 727
 - Receiver Class property, 345, 727
 - Receiver data, events, 320
 - ReceiverData property, 320
 - Recycle Bin command, 74
 - Recycle Bin setting, 81

- Recycle IIS Application Pool deployment step, 752
 - Recycle() method
 - SPFile object, 214
 - SPFolder object, 210
 - SPList object, 195
 - SPListItem object, 203
 - Recycling
 - files, 214
 - list items, 203
 - lists, 195
 - Redirect URL, events, 320
 - Redirect() method, 228
 - RedirectUrl property, 320
 - Refresh button, adding, 252
 - Refresh Button function, 252
 - Refresh Data function, 252
 - Refresh interval, setting, 252
 - Refresh Interval function, 252
 - Refreshing data display, 252
 - Regional Settings, 69
 - Regional settings, 189
 - RegionalSettings property, 189
 - @Register directive, 680
 - RelativeWebUrl property
 - SPItemEventProperties object, 325
 - SPWorkflowEventProperties object, 329
 - Release assemblies, empty projects, 114
 - Relevant Documents web parts, 50
 - Rendering
 - columns, lists, 218
 - fields, lists, 218
 - Reordering
 - folders, list items, 208
 - list items, 208
 - list items, user arrangement, 196
 - ReorderItems() method, 208
 - Replicator workflow activity, 531
 - Require content approval... setting, 246
 - Required Data Source Elements, 370
 - Required property
 - Data Elements, 370
 - SPField object, 217
 - Requires Resources property, 727
 - Reservations content type, 469
 - Reserving resources, 469
 - Reset to Site Definition action, 72
 - Reset Web Server, 746
 - Reset Web Server Mode On Upgrade, 746
 - ResolvePrincipals() method, 228
 - Resource content type, 468
 - Resource deployment type, 723, 724–725
 - Resource Group content type, 468
 - Resources
 - adding, 468
 - groups, adding, 468
 - reserving, 469
 - Retract Solution deployment step, 752
 - Retracting
 - solutions, 720
 - Visual Studio projects, 751–752, 773
 - Return Parameter Name property, 396
 - Return Type Descriptor property, 396
 - Reusable workflows, creating, 563–567
 - Reusing column definitions, 22, 24, 26
 - Revert XSLT function, 254
 - Ribbons, customizing, 33
 - Role assignments, returning, 195
 - Role inheritance, setting
 - list items, 203
 - lists, 196
 - RoleAssignments property, 195
 - RollbackTask workflow activity, 534
 - Rolling back tasks, 534
 - Root folder, returning, 188, 195
 - Root URL, specifying, 78
 - RootFile deployment type, 723
 - RootFinder property, 451
 - RootFolder property
 - SPList object, 195
 - SPWeb object, 188
 - RowLimit property
 - SPQuery object, 221
 - SPSiteDataQuery object, 224
 - RSS feeds, enabling, 80
 - RSS setting, 69
 - RSS Settings, 80
 - RuleSets, storing, 532
 - Run Post-Deployment Command deployment step, 752
 - Run Pre-Deployment Command deployment step, 752
- S**
- Safe Control Entries, 138
 - Sample data, generating, 253
 - Sample Data function, 253
 - Sandboxed solutions
 - available code elements, determining, 161–162
 - configuring, 87–88
 - CreateChildControls method, 167, 173–174
 - creating, 163
 - ItemAdding event handler, 161–162
 - ItemDeleting event handler, 168
 - Label1_Load handler, 164

- List Item event receiver, adding, 167–168
- Server object model, 160–162
 - web parts, adding, 166–167
- Save Site As Template action, 72–73
- Schedule and Reservations content type, 469
- Schedule content type, 469
- Scheduling appointments, 469
- Schema XML, getting and setting, 218
- Schema.xml file, modifying, 276–278
- SchemaXml property, 218
- Scope. *See also* SharePoint elements, scopes.
 - customization for visual web parts, 603
 - event handling, 530
 - Farm features, 700–701
 - features, 700–701
 - multiple, 730
 - returning, 218
 - Site features, 700–701
 - transactions, workflow activities, 531–532
 - view, specifying, 252
 - Web Application features, 700–701
 - Web features, 700–701
- Scope property
 - SPField object, 218
 - Visual Studio feature designer, 727
- Sealed property, 218
- Sealing columns and fields, 218
- Search Offline Availability setting, 69
- Search results, 28
- Search Server setting, 79
- Secure Store Service, 367
- Security. *See also* Permissions.
 - application pages, 692–696
 - settings, external content, 380–381
 - user ID, returning, 226
 - web applications, 656–657
 - web parts, configuring, 93
- Security command group
 - antivirus scanning, 92
 - Approve or Reject Distribution Groups command, 92
 - authentication providers, specifying, 92
 - Configure Information Rights Management, 93
 - Configure Managed Accounts, 92
 - Configure Password Change Settings, 92
 - Configure Self-Service Site Creation, 93
 - Configure Service Accounts, 92
 - Define Blocked File Types, 92
 - distribution groups, approving or rejecting, 92
 - farm administrators, managing, 92
 - file types, blocking, 92
 - Information Rights Management, enabling, 93
 - Manage Antivirus Settings, 92
 - Manage the Farm Administrators Group command, 92
 - Manage Web Part Security, 93
 - managed accounts, configuring, 92
 - passwords, changing, 92
 - self-service site creation, enabling, 93
 - service accounts, configuring, 92
 - Specify Authentication Providers, 92
 - web part edits, enabling, 93
 - web part security, configuring, 93
- Security Configuration setting, 78
- Security headers, adding to documents, 80
- Security policies, resolving principals, 228
- Security validation expiration settings, 81
- SecurityBits attribute, 275
- SecurityPermissions, 653
- Selectable in Clients permission, 381
- Self-service site creation, enabling, 85, 93
- Self-Service Site Creation command, 83
- Send User Name and Password in E-Mail setting, 81
- SendActivity workflow activity, 532
- SendEmail workflow activity, 534
- SendEmail() method, 228
- Sequence attribute, 275
- Sequence Number subelement, 132
- Sequence workflow activity, 531
- Sequential workflow execution, 531
- Sequential Workflow project items, 124
- Sequential Workflow projects, 108, 109
- Sequential workflows, 526, 562
- Server Explorer
 - content type properties, displaying, 95–97
 - ContentTypes subnode, 95–97
 - Document Libraries subnode, 98
 - features, displaying, 98, 99
 - Features subnode, 98, 99
 - galleries, displaying, 98, 101
 - lists, displaying, 98, 101
 - Lists and Libraries subnode, 98, 101
 - ListTemplates subnode, 98, 100
 - Shared Documents library, displaying, 98, 101
 - templates, listing, 98, 100
 - workflow associations, displaying, 101–102
 - Workflow Associations subnode, 101–102
 - workflow templates, displaying, 101–102
 - Workflow Templates subnode, 101–102
- Server object model, 160–162

- ServerRelative Url property
 - SPWeb object, 189
 - SPWebEventProperties object, 328
- Server-relative URLs
 - getting and setting, 189
 - returning, 214
- ServerRelativeUrl property
 - SPFile object, 214
 - SPFolder object, 211
- Servers
 - administering. *See* Central Administration page.
 - e-mail. *See* SMTP servers.
 - farm, managing, 87
 - keeping up to date, 92
- Servers property, 177
- Server-side object model
 - creating lists programmatically, 284, 285–294
 - modifying lists programmatically, 300–304
- Service accounts, configuring, 92
- Service Application Connections setting, 79
- Service applications, 85
- Service Applications, specifying, 82
- Service Applications command group, 85
- Service Connections command, 82
- Services property, 177
- Set Permissions permission, 381
- SetState workflow activity, 534
- Shared Documents library, displaying, 98, 101
- SharePoint
 - content database, specifying, 78
 - installing. *See* Installing SharePoint.
- SharePoint Designer
 - installing, 807–808
 - overview, 7
- SharePoint Designer Settings, 74, 82
- SharePoint elements. *See also* XML elements.
- SharePoint elements, and project items
 - Application Pages, 708
 - Content Type, 707
 - Content Type Binding, 707
 - Custom Action, 707
 - Custom Action Group, 707
 - Delegate Controls, 707
 - Document Converter, 707
 - Event Receiver, 707
 - Feature/Site Template Association, 707
 - Field Element, 707
 - Hide Custom Action, 707
 - List Instance, 707
 - List Template, 707
 - Module, 707
 - for Visual Studio 2010, 708
 - Visual Web Parts, 708
 - Workflow, 707
- SharePoint elements, scopes
 - Content Type, 709
 - Content Type Binding, 709
 - Custom Action, 709
 - Custom Action Group, 709
 - Delegate Controls, 709
 - Document Converter, 709
 - Event Receiver, 709
 - Feature/Site Template Association, 709
 - Field Element, 709
 - Hide Custom Action, 709
 - List Instance, 709
 - List Template, 709
 - Module, 709
 - multiple scopes, 730
 - Workflow, 709
- SharePoint Events property, 138–139
- SharePoint feature management, 82
- SharePoint pages. *See also* Application pages; Master pages; Site pages.
 - look and feel. *See* Master pages; User interface.
 - overview, 655
- SharePoint pages, architecture
 - application pages, 660–661
 - master pages, 658–659
 - page types, 658–661
 - site collections, 657
 - site pages, 659–660
 - web applications, 656–657
- SharePoint pages, master pages
 - creating, 466
 - default, specifying, 81
 - URL, returning, 189
- SharePoint Product Version, 747
- SharePoint property, 129–130, 137
- SharePoint servers, administering. *See* Central Administration page.
- SharePoint workflows. *See* Workflow activities, SharePoint-specific; Workflows, SharePoint.
- SharePointProjectItem.spdata file, 543
- SharePointSecurityPermission, 653
- Show In Picker property, 370
- ShowInEditForm property, 218
- ShowInSearchUI property, 451
- Sid property, 226
- Sign In Page URL setting, 78
- Significant property, 394
- Sign-in page, specifying, 78

- Silverlight
 - debugging, 779–786
 - web parts. *See* Web parts, Silverlight.
- Site actions. *See* Site settings, site actions.
- Site associations, workflows, 548–552
- Site changes, playing back, 5
- Site Collection Features command, 74
- Site Collection Web Analytics Report action, 72
- Site collections. *See also* SPSite object.
 - adding, 85
 - administering. *See* Site settings, site collection administration.
 - administrators, configuring, 85
 - configuring, 85
 - creating, 8, 85, 809–811
 - definition, 8, 657
 - deleting, 85
 - locking, 85
 - object models, 154
 - quotas for defaults, 80
 - self-service site creation, enabling, 85
 - settings. *See* Site settings.
 - vs.* sites, 657
 - viewing, 85
- Site Collections command group, 85
- Site columns. *See also* List columns, site columns.
 - adding to default library view, 504–506
 - additional settings, 461
 - column data types, 460
 - column group name, 460–461
 - column name, 460
 - components of, 459–461
 - creating, 461–464, 481–483
 - predefined column groups, 461
- Site Columns setting, 68
- Site content. *See* Libraries; Lists; Pages; Subsites (workspaces); Workflows.
- Site Content Types setting, 68
- Site Definition projects, 109
- Site Hierarchy command, 74
- Site Libraries and Lists setting, 69
- Site pages
 - architecture, 659–660
 - ASPX pages, customizing, 674
 - controls, registering, 680
 - creating, 672–682
 - customizing, 673–682
 - deploying, 682
 - deployment location, specifying, 682
 - HTML pages, customizing, 674
 - inline code, 684
 - @Page directive, 679
 - @Register directive, 680
 - in Visual Studio projects, 676–682
- Site permissions, 66
- Site property
 - SPContext object, 172
 - SPWeb object, 187
- Site scope, features, 700–701
- Site settings
 - child sites *vs.* top-level, 57–58, 59
 - overview, 57–58
- Site settings, galleries
 - customizing, 68
 - definition, 67
 - list of, 68
 - List Templates, 68
 - Master Pages, 68
 - Site Columns, 68
 - Site Content Types, 68
 - Solutions, 68
 - Themes, 68
 - Web Parts, 68
- Site settings, look and feel
 - colors, 70
 - definition, 69
 - fonts, 70
 - list of, 70
 - Quick Launch, 70–71
 - Site Theme, 70
 - Title, Description, and Icon, 70
 - Top Link Bar, 70
 - Tree View, 70
- Site settings, site actions
 - definition, 71
 - Delete This Site, 72
 - Manage Site Features, 72
 - managing features, 71–72
 - Reset to Site Definition, 72
 - Save Site As Template, 72–73
 - Site Collection Web Analytics Report, 72
 - Site Web Analytics Report, 72
 - stsadmin.exe tool, 72
- Site settings, site collection administration
 - commands, 74
 - deleted items, restoring, 74
 - description, 73
 - Help Settings, 74
 - Portal Site Connection command, 74
 - Recycle Bin command, 74
 - SharePoint Designer Settings, 74
 - Site Collection Features command, 74
 - Site Hierarchy command, 74
 - user interface version, specifying, 74
 - Visual Upgrade, 74

- Site settings, system administration
 - definition, 69
 - list of, 69
 - Regional Settings, 69
 - RSS, 69
 - Search Offline Availability, 69
 - Site Libraries and Lists, 69
 - Sites and Workspaces, 69
 - User Alerts, 69
 - Workflow Settings, 69
 - Workflows, 69
- Site settings, users and permissions
 - group management, 60–61
 - overview, 58, 60
 - permission levels
 - Add and Customize Pages, 66
 - Add Items, 65
 - Add/Remove Personal Web Parts, 67
 - Apply Themes and Borders, 66
 - Approve Items, 65
 - Browse Directories, 66
 - Browse User Information, 66
 - Create Alerts, 65
 - Create Groups, 66
 - Create Subsites, 66
 - creating your own, 64
 - Delete Items, 65
 - Delete Versions, 65
 - Edit Items, 65
 - Edit Personal Information, 67
 - editing, 63–64
 - Enumerate Permissions, 66
 - granting, 63
 - group settings, 60–62
 - inheriting, 62
 - list permissions, 65
 - Manage Alerts, 67
 - Manage Lists, 65
 - Manage permissions, 66
 - Manage Personal Views, 67
 - Manage Web Site, 66
 - Open Items, 65, 67
 - Override Checkout, 65
 - personal permissions, 66
 - site permissions, 66
 - Update Personal Web Parts, 67
 - Use Client Integration Features, 67
 - Use Self-Service Site Creation, 66
 - User Remote Interfaces, 67
 - View Application Pages, 65
 - View Items, 65
 - View Pages, 66
 - View Versions, 65
 - View Web Analytics Data, 66
- Site Theme setting, 70
- Site Users web parts, 50
- Site Web Analytics Report action, 72
- SiteId property, 320
- Site-relative path, returning
 - columns, lists, 218
 - fields, lists, 218
- Site-relative URLs, returning
 - files, 213
 - list items, 201
 - local path, 228
- Sites
 - active features, returning, 187
 - alerts, returning, 188
 - auditing, 188
 - author, returning, 188
 - child sites, returning, 188
 - closing, 188
 - creation timestamp, returning, 187
 - currency settings, regional, 189
 - current user, returning, 189
 - deleting, 187
 - description, getting and setting, 187
 - existence, verifying, 188
 - exploring with Visual Studio. *See* Server Explorer.
 - files, returning, 187
 - GUID, returning, 187
 - ID, returning, 187
 - key value pairs for settings, returning, 187
 - language locale, returning, 188
 - lists, querying, 189
 - lists, returning, 187
 - look and feel. *See* Site settings, look and feel.
 - master page URL, returning, 189
 - navigation, 188
 - numeric field settings, regional, 189
 - object models, 154
 - parent site, returning, 188
 - parent site collection, returning, 187
 - provisioning status, returning, 189
 - quick launch area, accessing, 188
 - regional settings, 189
 - root folder, returning, 188
 - server-relative URLs, getting and setting, 189
 - vs.* site collections, 657
 - site definition, returning, 189
 - site template, returning, 189
 - subfolders, returning, 188
 - theme, returning, 188
 - time settings, regional, 189
 - titles, getting and setting, 187

- unsafe updates, allowing, 188
- updating, 187
- URL, returning, 187
- users, adding, 189
- workflow instances, returning, 188
- Sites, creating
 - child sites, 8–12, 13
 - Create dialog, 10–11, 13
 - during installation, 8
 - self-service site creation, enabling, 93
 - site name, specifying, 10–11
 - templates, choosing, 10
 - templates, creating your own, 12
 - URL, specifying, 10–11
 - user-created sites, enabling, 83
- Sites and Workspaces setting, 69
- Sites property, 179
- SiteTemplates folder, deploying projects to, 120
- SMTP servers, configuring, 87
- Social Meeting Workspace subsites, 54
- Solution Id property, 727
- Solution packaging
 - manifest.xml file, 713–714
 - overview, 712–713
- Solution packaging, solution manifest file overview, 713–714
- XML elements
 - ActivationDependencies, 715
 - ApplicationResourceFiles, 715
 - Assemblies, 715
 - ClassResources, 716
 - CodeAccessSecurity, 716
 - DwpFiles, 716
 - FeatureManifests, 716
 - Resources, 716
 - RootFiles, 716
 - SafeControls, 715
 - SiteDefinitionManifests, 717
 - TemplateFiles, 717
 - WebTempFile, 717
- Solutions. *See also* Projects.
 - deploying, 717–720
 - reviewing, 90
 - sandbox, configuring, 87–88
- Solutions property, 177
- Solutions setting, 68
- Sort & Group function, 251
- Sort setting, 39
- Sorting
 - document columns, 39
 - items, custom list views, 251
- Source entity, 443
- Source ID identifier, 443
- Source View *vs.* Code View, 589
- Spaces, in file paths, 353
- SPContext object
 - definition, 154
 - event receivers, 175–176
 - properties and methods, 172–173, 174–175. *See also specific properties and methods.*
 - returning, 172
- Specific Finder method, creating, 438–440
- Specific Finder method instance type, 395
- Specify Authentication Providers, 92
- SPEmailEventReceiver, 327
- SPException object
 - definition, 155
 - properties and methods, 227–228. *See also specific properties and methods.*
- SPFarm object
 - definition, 155
 - properties and methods, 177–179. *See also specific properties and methods.*
 - returning, 177
- SPFeatureReceiverProperties object
 - Definition property, 331
 - Feature property, 331
 - UserCodeSite property, 331
- SPField object
 - definition, 155
 - properties and methods, 217–220. *See also specific properties and methods.*
- SPFieldCollection object, returning, 173
- SPFile object
 - definition, 155
 - properties and methods, 212–216. *See also specific properties and methods.*
 - returning, 173
- SPFolder object. *See also* Folders, lists.
 - definition, 155
 - properties and methods, 210–212. *See also specific properties and methods.*
- SPItem object, returning, 173
- SPItemEventProperties object
 - AfterProperties property, 324, 325–326
 - AfterUrl property, 324
 - BeforeProperties property, 324, 325–326
 - BeforeUrl property, 324
 - Context property, 324
 - CurrentUserid property, 324
 - FileSystemObjectTypePropertyName property, 324
 - List property, 324
 - ListId property, 324
 - ListItem property, 325
 - ListItemId property, 325

- SPItemEventProperties object, *continued*
 - ListTitle property, 325
 - RelativeWebUrl property, 325
 - UserDisplayName property, 325
 - UserLoginName property, 325
 - Versionless property, 325
 - WebUrl property, 325
 - Zone property, 325
- SPItemEventReceiver, 322–327
- SPLimitedWebPartManager object, returning, 214
- SPLINQ (LINQ to SharePoint), querying lists, 307–315
- SPList item instance, 325
- SPList object
 - definition, 154
 - properties and methods, 194–201. *See also specific properties and methods.*
 - returning, 172
- SPListEventProperties object
 - FeatureId property, 321
 - Field property, 321
 - FieldName property, 321
 - FieldXml property, 322
 - List property, 322
 - ListId property, 322
 - ListTitle property, 322
 - TemplateId property, 322
 - UserDisplayName property, 322
 - UserLoginName property, 322
 - WebId property, 322
 - WebUrl property, 322
- SPListEventReceiver, 321–322
- SPListItem object
 - definition, 154
 - properties and methods, 201–206. *See also specific properties and methods.*
 - returning, 173, 194, 213
- SPListItemCollection object
 - definition, 155
 - properties and methods, 206–209. *See also specific properties and methods.*
- SPMetal.exe tool, querying lists, 308–309
- SPMobileBaseFieldControl class, returning, 218
- SPPrincipalInfo object, resolving principals, 228
- SPQuery object. *See also* SPSiteDataQuery object.
 - definition, 155
 - properties and methods, 221–223. *See also specific properties and methods.*
- SPRequest object, 155
- SPUrlZone enumeration, 325
- SPSecurity object, 155
- SPServerCollection object, returning, 177
- SPServiceCollection object, returning, 177
- SPSite, 157–158
- SPSite object. *See also* Site collections.
 - definition, 154
 - disposing, 690
 - disposing of, 157, 174
 - returning, 172
 - vs.* SPWeb object, 173
- SPSiteDataQuery object. *See also* SPQuery object.
 - definition, 156
 - properties and methods, 223–226. *See also specific properties and methods.*
- SPSolutionCollection object, returning, 177
- SPUser object. *See also* Users.
 - definition, 154
 - properties and methods, 226–227. *See also specific properties and methods.*
- SPUtility object
 - definition, 156
 - methods, 228–229
- SPWeb object. *See also* Sites.
 - definition, 154
 - disposing of, 157, 174
 - properties and methods, 187–193. *See also specific properties and methods.*
 - returning, 172
 - vs.* SPSite object, 173
- SPWebApplication object
 - definition, 155
 - properties and methods, 179–181. *See also specific properties and methods.*
- SPWebEventProperties object
 - FullUrl property, 328
 - NewServerRelativeUrl property, 328
 - ParentWebId property, 328
 - ServerRelativeUrl property, 328
 - UserDisplayName property, 328
 - UserLoginName property, 328
 - Web property, 328
 - WebId property, 328
- SPWebEventReceiver, 327–328
- SPWorkflowEventProperties object
 - ActivationProperties property, 329
 - AssociationData property, 329
 - CompletionType property, 329
 - ErrorException property, 329
 - InitiationData property, 329
 - InstanceId property, 329
 - PostponedEvent property, 329
 - RelativeWebUrl property, 329

- TerminatedByUserId property, 330
 - WebUrl property, 330
 - SPWorkflowEventReceiver, 328–330
 - SQL Server connection, 367–368
 - SQL Server external source type, 366
 - SSL, enabling, 78
 - Stack traces, returning, 228
 - Standard View
 - description, 16
 - illustration, 17
 - reverting to a previous version, 25
 - tracking changes, 25–26
 - Standard View, creating
 - attachments, 27
 - choosing a format, 16, 19, 20
 - content types setting, 27
 - datasheet view editing, enabling, 28
 - dialogs, enabling, 28
 - folders, 28
 - item-level permissions, 27
 - naming the view, 19
 - offline client availability, 28
 - permissions, 28–30
 - search results, customizing, 28
 - Standard View, creating columns
 - arranging, 19
 - Create Column dialog, 23, 24
 - custom site columns, 27
 - List tab, 22
 - reusing column definitions, 22, 24, 26
 - from scratch, 22
 - selecting, 19
 - settings, illustration, 21
 - Standard workflow activities, 529
 - Startup settings, empty projects, 114
 - State Machine Workflow project items, 124
 - State Machine Workflow projects, 108, 109
 - State machine workflows, 527–528, 534
 - Static web part connections, 637
 - StaticName property, 218
 - Status property, 320
 - Strongly-typed queries, 307–315
 - stsadmin.exe tool, 72, 348
 - Style setting, 40
 - Styles, lists, 40
 - Subelements, Elements.xml file, 132
 - SubFolders property, 210
 - Subsites (workspaces)
 - creating, 55–57
 - definition, 54
 - Subsites (workspaces), types of
 - Basic Meeting Workspace, 54
 - Blank Meeting Workspace, 54
 - Blank Site, 54
 - Blog, 54
 - Decision Meeting Workspace, 54
 - Document Workspace, 54
 - Group Work Site, 54
 - Multipage Meeting Workspace, 54, 56–57
 - Social Meeting Workspace, 54
 - Team Site, 54
 - Summary Task content type, 467
 - Survey, list template, 234
 - Survey lists, 15
 - Suspend workflow activity, 532
 - Suspending workflow execution, 532
 - SynchronizationScope workflow activity, 532
 - Synchronizing workflow execution, 532
 - Synchronous (before) events, 318
 - System administration. *See* Central Administration page; Site settings, system administration.
 - System settings, 86–88. *See also* Site settings, system administration.
 - SystemSecurityPermission, 653
 - SystemUpdate() method, 203
- ## T
- Table tab, 252–253
 - Tabular View setting, 40
 - Tabular views, 40
 - Task content type, 469
 - Task list type, 364
 - Task lists, 15
 - Task-related workflow activities, 532
 - Tasks
 - change events, 533
 - creating, 469, 533
 - creating with content type, 533
 - creation events, 533
 - deleting, 533
 - deletion events, 533
 - executing, workflow activities, 529
 - marking complete, 533
 - rolling back, 534
 - updating, 534
 - in workflows, creating, 543–545, 568–569
 - in workflows, referencing, 546
 - Tasks, list template, 234
 - Tasks property, 202
 - Team Site Members permission level, 28–29
 - Team Site Owners permission level, 28–29
 - Team Site subsites, 54
 - Team Site Visitors permission level, 28–29
 - TEMPLATE folder, deploying projects to, 120
 - TemplateFile deployment type, 723

- TemplateId property, 322
- Templates
 - from custom list views, 255–261
 - Default Quota Template setting, 80
 - list instances, 238–241
 - List Templates setting, 68
 - listing, 98, 100
 - lists, 489–490
 - for lists. *See* List templates.
 - ListTemplates subnode, 98, 100
 - name, specifying, 275
 - ordering priority, 275
 - quota, modifying, 85
 - Save Site As Template action, 72–73
 - saving from lists, 28
 - saving lists as, 28
 - sites, 10, 12
 - templates, listing, 98, 100
 - type, specifying, 279
 - Word documents, 494–497
 - workflow, displaying, 101–102
 - Workflow Templates subnode, 101–102
- TemplateType attribute, 279
- Terminate workflow activity, 531
- TerminatedByUserId property, 330
- Testing
 - event receivers, 150
 - Silverlight web parts, 634–636
 - Visual Studio deployments, 765
 - visual web parts, 607
 - workflows, 569–570
- Text boxes, inserting, 254
- Text Message Service Settings, 82
- Textbox controls, visual web parts, 587–589
- Theme property, 188
- Themes setting, 68
- Throttling HTTP requests, 81
- Throw workflow activity, 532
- Time. *See* Date and time.
- Time stamp of last update, 370
- Time zone, setting, 80
- Timecard content type, 468
- Timecard data, adding, 468
- Timer jobs, 88–91
- Timer Jobs command group, 90
- Timestamp Field property, 370
- Timestamp filter, 374
- Title, Description, and Icon setting, 70
- Title attribute, 279
- Title property
 - SPField object, 217
 - SPFile object, 213
 - SPList object, 194
 - SPListItem object, 202
 - SPWeb object, 187
 - Visual Studio feature designer, 728
- Titles
 - lists, getting and setting, 194
 - sites, getting and setting, 187
 - visual web parts, changing, 591–592
 - web parts, setting, 254
- Titles, returning
 - columns, lists, 217
 - fields, lists, 217
 - files, 213
 - list items, 202
- To File function, 255
- To Site Gallery function, 255
- Tokens, 335
- Toolbar options, custom list views, 253
- Tooltip description, visual web parts, 603
- Top Link Bar setting, 70
- Totals row, displaying, 253
- Totals Row function, 253
- Totals setting, 40
- Tracking
 - changes, 25–26
 - issues, 469
- Transaction scope, workflow activities, 531–532
- TransactionScope workflow activity, 532
- Tree View setting, 70
- Triggering events
 - user display name, 325, 328
 - user ID, 324
 - user login name, 325
 - web events, 328
- Triggers for events, 319–320
- Trust levels, web part configuration, 650
- Trusted entities, identifying, 138
- Try, catch, finally pattern, 159
- Type attribute, Elements.xml file, 275
- Type descriptor, Visual Studio external content types
 - copying, 414–417
 - defining, 409–414, 435–438
 - manually adjusting, 421
 - name, setting, 434–435
- Type descriptor properties, Visual Studio external content types
 - Associated Filter, 393
 - Creator Field, 393
 - Custom Properties, 393
 - Default Display Name, 393
 - Foreign Identifier Association Entity, 393
 - Foreign Identifier Association Name, 393
 - Identifier, 393
 - Identifier Entity, 393
 - Is Cached, 394

- Is Collection, 394
 - LOB Name, 394
 - Name, 394
 - Pre-Updater Field, 394
 - Read-Only, 394
 - Significant, 394
 - Type Name, 394
 - Update Field, 394
 - Type descriptors, Visual Studio external
 - content types
 - caching, 394
 - creating a new ECT, 393
 - custom properties, 393
 - data structure name, 394
 - display name, default, 393
 - filters, setting, 393
 - foreign identifier association entity, 393
 - foreign identifier association name, 393
 - identifier, 393
 - identifier entity, 393
 - name, 394
 - .NET type, 394
 - read-only, 394
 - representing collections, 394
 - significant values, 394
 - Update Field, 394
 - updating, 394
 - Type Name property
 - setting, 408
 - type descriptors, 394
 - Type property
 - method instance, Visual Studio external
 - content types, 396
 - SPField object, 217
 - Type subelement, Elements.xml file, 132
 - TypeDisplayName property, 218
- U**
- UIVersion property, 728
 - Uninstalling
 - features, 711
 - Visual Studio deployments, 773
 - UniqueId property, 203
 - Unsafe updates, allowing
 - sites, 188
 - visual web parts, 606–607
 - Update Field property, 394
 - Update Personal Web Parts permission
 - level, 67
 - UpdateAllTasks workflow activity, 534
 - UpdateItemActivity workflow activity, 535
 - Update() method
 - creating, 417–420
 - SPFarm object, 177
 - SPField object, 217
 - SPFile object, 213
 - SPFolder object, 210
 - SPList object, 194
 - SPListItem object, 202
 - SPUser object, 226
 - SPWeb object, 187
 - SPWebApplication object, 179
 - Updater() method
 - creating, 438–440
 - instance type, 395
 - UpdateTask workflow activity, 534
 - Updating
 - columns, lists, 217
 - entity objects, Visual Studio external content types, 395
 - farms, 177
 - fields, lists, 217
 - files, 213
 - list items, 202, 203
 - properties, 447–448
 - sites, 187
 - tasks, 534
 - users, 226
 - web applications, 179
 - workflow items, 535
 - Upgrade Actions Receiver Assembly property, 728
 - Upgrade Actions Receiver Class property, 728
 - Upgrade and Patch Management command
 - group, 92
 - UpgradeActions, XML element, 704–706
 - Upload limits, specifying, 81
 - Url attribute, Elements.xml file, 279
 - Url property
 - SPFile object, 213
 - SPFolder object, 210
 - SPListItem object, 201
 - SPWeb object, 187
 - URLs
 - deploying projects to, 115
 - site-relative. *See* Site-relative URLs.
 - Use Client Integration Features permission
 - level, 67
 - Use Self-Service Site Creation permission
 - level, 66
 - UseClientCachingForSearch property, 451
 - User accounts, setting, 83
 - User Alerts setting, 69
 - User Control project items, 124
 - User interface. *See also* Site settings, look and feel.
 - components, 446
 - version, specifying, 74

- User interface, creating
 - external content type, SharePoint Designer, 376–380
 - forms, 376–380
 - lists, 376–380
 - User names, including in e-mail, 81
 - User Permissions command, 83
 - User Policy command, 83
 - User Remote Interfaces permission level, 67
 - User Tasks web parts, 50
 - User tokens, returning, 226
 - UserCodeSite property, 331
 - User-created sites, enabling, 83
 - User-Defined Workflows setting, 82
 - UserDisplayName property
 - SPItemEventProperties object, 325
 - SPListEventProperties object, 322
 - SPWebEventProperties object, 328
 - UserLoginName property
 - SPItemEventProperties object, 325
 - SPListEventProperties object, 322
 - SPWebEventProperties object, 328
 - Users. *See also* SPUser object.
 - adding to sites, 189
 - alerts, returning, 226
 - authentication, 226
 - e-mail address, getting and setting, 226
 - group information, returning, 226
 - group work, adding, 468
 - int ID, returning, 226
 - login name, returning, 226
 - names, getting and setting, 226
 - notes, getting and setting, 226
 - object models, 154
 - security ID, returning, 226
 - updating, 226
 - user tokens, returning, 226
 - XML representation, 226
 - Users content type, 468
 - UserToken property, 226
 - using directives, adding, 590–591
 - using keyword, disposing of objects, 158
 - Utilities. *See* SPUtility object.
 - Utility class, object models, 156
- V**
- V4.master page, 661, 664
 - Version attribute, 473
 - Version property
 - SPField object, 217
 - SPList object, 194
 - Visual Studio feature designer, 728
 - Versioning settings, 25–26
 - Versionless property, 325
 - VersionRange, XML element, 705
 - Versions
 - content types, Visual Studio, 473
 - history, returning, 202, 213
 - returning, 194, 217
 - reverting to a previous version, 25
 - tracking, 325
 - Versions property
 - SPFile object, 213
 - SPListItem object, 202
 - View Application Pages permission level, 65
 - View Attributes property, 221
 - View Items permission level, 65
 - View Pages permission level, 66
 - View scope, specifying, 252
 - View style, specifying, 253
 - View Style function, 253
 - View Versions permission level, 65
 - View Web Analytics Data permission level, 66
 - ViewFields property
 - SPQuery object, 221
 - SPSiteDataQuery object, 224
 - Viewing
 - list items, 31–32
 - and managing external settings, 364
 - results, 382–384
 - site collections, 85
 - Views, documents
 - columns, 39
 - filtering, 39
 - folders, 40
 - group by, 40
 - inline editing, 40
 - item limit, 40
 - mobile, 40
 - sorted, 39
 - style, 40
 - tabular, 40
 - totals, 40
 - Views, lists, 16
 - Views property, 195
 - Visibility, getting and setting
 - columns, lists, 217
 - fields, lists, 217
 - lists, 195
 - Visual Studio
 - administrator privileges, 93–94
 - exploring SharePoint sites. *See* Server Explorer.
 - installing, 803–807
 - launching, 93
 - overview, 5–7
 - Visual Studio 2010 SDK, installing, 760

- Visual Studio projects
 - overview, 721–725
 - Packaging Explorer, 747–748
- Visual Studio projects, configurable deployment. *See also* Visual Studio projects, deployment.
 - Activate Features step, 752
 - activating configurations, 759–760
 - Add Solution step, 752
 - auto-retract, 779
 - configurations, changing, 753–759
 - conflicts, 774–777
 - default configurations, 751
 - default deployment configuration, 751, 753–754
 - farm solutions, 777–778
 - no activation deployment, 751, 753–754
 - overview, 751
 - Recycle IIS Application Pool step, 752
 - Retract Solution step, 752
 - retraction, 751–752
 - Run Post-Deployment Command step, 752
 - Run Pre-Deployment Command step, 752
 - sandboxed solutions, 777–778
 - SharePoint elements, conflict resolution
 - Content Type, 776
 - List Instance, 776
 - Module, 776
 - Visual Web Part, 776
 - Web Part, 776
 - SharePoint elements, sandboxed solutions
 - Content Types, 778
 - Custom Actions, 778
 - Event Receivers, 778
 - Fields, 778
 - List Definitions, 778
 - List Instances, 778
 - Module, 778
 - Web Parts, 778
 - Silverlight, debugging, 779–786
 - steps, custom
 - CanExecute() method, 761–765
 - changing to a farm solution, 770–771
 - class discovery, 771–773
 - custom class, implementing, 764–765
 - definition, 761
 - deployment, 765, 773
 - Execute() method, 761–765
 - inheritance, 771
 - Initialize() method, 761–765
 - installation, 766–770
 - installing Visual Studio 2010 SDK, 760
 - package information, specifying, 765–766
 - references, adding to assemblies, 760
 - retraction, 773
 - testing, 765
 - uninstalling, 773
 - steps, standard, 752
- Visual Studio projects, deployment. *See also* Visual Studio projects, configurable deployment.
 - adding assemblies, 748–751
 - deploying, 722–725
 - Deployment Location property, 722–725
 - Deployment Type property, 722–725
- Visual Studio projects, deployment types
 - AppGlobalResource, 723, 724–725
 - ApplicationResource, 723, 724
 - ClassResource, 723, 724
 - DwpFile, 723
 - ElementManifest, 723
 - NoDeployment, 723, 724
 - PackageOnly, 723
 - Resource, 723, 724–725
 - RootFile, 723
 - TemplateFile, 723
- Visual Studio projects, feature designer
 - creating an empty project, 728–729
 - custom action elements, creating, 731–732
 - custom action elements, modifying, 732–733
 - custom properties, creating, 743–745
 - empty elements, 731
 - feature activation dependencies, 739–743
 - mapped folders, 733–739
 - overview, 725–726
 - project items, adding, 730
 - properties
 - Active On Default, 727
 - Always Force Install, 727
 - Auto Activate In Central Admin, 727
 - Creator, 727
 - Default Resource File, 727
 - Deployment Path, 727
 - Description, 727
 - Feature Id, 727
 - Image Alt Text, 727
 - Image Url, 727
 - Is Hidden, 727
 - Receiver Assembly, 727
 - Receiver Class, 727
 - Requires Resources, 727
 - Scope, 727
 - Solution Id, 727
 - Title, 728

- Visual Studio projects, feature designer
 - properties, *continued*
 - UIVersion, 728
 - Upgrade Actions Receiver Assembly, 728
 - Upgrade Actions Receiver Class, 728
 - Version, 728
 - SharePoint elements, multiple scopes, 730
 - Visual Studio projects, package designer
 - deployment, adding assemblies, 748–751
 - overview, 745–746
 - properties
 - Deployment Server Type, 746
 - Description, 746
 - Name, 746
 - Reset Web Server, 746
 - Reset Web Server Mode On Upgrade, 746
 - SharePoint Product Version, 747
 - Visual Studio Tools for Office (VSTO), 512–520
 - Visual Upgrade, 74
 - Visual Web Part
 - adding to farm solutions, 163–164
 - project items, 124
 - projects, 107, 109
 - SharePoint element, conflict resolution, 776
 - Visual web parts
 - creating with AJAX, 586, 597–600
 - customizing, 600–609
 - partial updating, enabling, 586, 597–600
 - Visual web parts, creating with visual web designer
 - basic project, 581–583
 - breakpoints, setting, 592
 - button controls, 589
 - controls, adding, 587–589
 - debugging the web part, 592–597
 - deploying as farm solutions, 583
 - Elements.xml file, 583
 - file structure, 583–586
 - headings, size and color, 586–587
 - Source View *vs.* Code View, 589
 - textbox controls, 587–589
 - titles, changing, 591–592
 - using directives, adding, 590–591
 - VisualWebPart1.cs file, 583
 - VisualWebPart1UserControl.ascx file, 583
 - VisualWebPart1.webpart file, 583
 - Visual web parts, custom properties
 - attributes, 603
 - category, defining, 603
 - Category attribute, 603
 - customization scope, setting, 603
 - editing, 605
 - editing view, 603, 604–605
 - implementing, 605–606
 - name, displaying, 603
 - Personalizable attribute, 603
 - testing, 607
 - tooltip description, 603
 - unsafe update, allowing, 606–607
 - WebBrowsable attribute, 603
 - WebDescription attribute, 603
 - WebDisplayName attribute, 603
 - Visual Web Parts, SharePoint element, 708
 - VisualWebPart1.cs file, 583
 - VisualWebPart1UserControl.ascx file, 583
 - VisualWebPart1.webpart file, 583
 - Vs. Visual Studio, 384–385
 - VSTO (Visual Studio Tools for Office), 512–520
- ## W
- WCF end-points, connection to, 532
 - WCF Service external source type, 366
 - WCF services, workflows as, 532
 - Web application directories, deploying
 - projects to, 118
 - Web Application scope, features, 700–701
 - Web applications. *See also* SPWebApplication object.
 - alerts, configuring, 80
 - Alerts setting, 80
 - anonymous access, enabling, 78, 83
 - Anonymous Policy command, 83
 - Application Pool setting, 78
 - authentication provider, specifying, 78
 - authentication providers, configuring, 83
 - Authentication Providers command, 83
 - Authentication setting, 78
 - Blocked File Types command, 83
 - Blog API Settings, 80
 - Browser File Handling setting, 80
 - change log expiration period, setting, 82
 - Change Log setting, 82
 - Claims Authentication Type setting, 78
 - Claims Based Authentication, 78
 - Classic Mode Authentication, 78
 - content databases, returning, 180
 - customer experience data, collecting, 79
 - Customer Experience Improvement Program, 79
 - Daily Time Window for Large Queries setting, 81
 - database large queries, specifying a window for, 81

- Database Name and Authentication setting, 78
- database operation limits, 81
- Default Quota Template setting, 80
- Default Time Zone setting, 80
- deleting, 179
- Extend command, 80
- Fallover Server setting, 79
- features, returning, 179
- file types, blocking, 83
- Forms Based Authentication, enabling, 78, 79
- host site, specifying, 78
- HS Web Site setting, 78
- HTTP Request Monitoring and Throttling, 81
- list permissions, specifying maximum number of, 81
- List Unique Permissions Threshold setting, 81
- List View Threshold setting, 81
- Mail Settings, 82
- making available to multiple IIS sites, 80
- Manage Features command, 82
- Managed Path command, 82
- master page default, specifying, 81
- Master Page Setting for
 - Application_Layout Pages, 81
- Maximum Upload Size setting, 81
- MetaWeblog API, enabling, 80
- monitoring HTTP requests, 81
- name, returning, 180
- online presence information, enabling, 80
- overview, 76–77
- passwords, including in e-mail, 81
- path management, specifying, 82
- permission policies, setting, 83
- Permission Policy command, 83
- policies, returning, 180
- Presence Settings, 80
- provisioning, 180
- Public URL setting, 78
- quotas for default site collections, 80
- Recycle Bin setting, 81
- root URL, specifying, 78
- RSS feeds, enabling, 80
- RSS Settings, 80
- Search Server setting, 79
- Security Configuration setting, 78
- security headers, adding to documents, 80
- security validation expiration settings, 81
- Self-Service Site Creation command, 83
- Send User Name and Password in E-Mail setting, 81

- Service Application Connections setting, 79
- Service Applications, specifying, 82
- Service Connections command, 82
- settings, accessing. *See* SPWebApplication object.
- settings and policies, 77–79
- SharePoint content database, specifying, 78
- SharePoint Designer Settings, 82
- SharePoint feature management, 82
- Sign In Page URL setting, 78
- sign-in page, specifying, 78
- site collections, returning, 179
- SSL, enabling, 78
- state, accessing. *See* SPWebApplication object.
- Text Message Service Settings, 82
- throttling HTTP requests, 81
- time zone, setting, 80
- updating, 179
- upload limits, specifying, 81
- Uri lookup, 179
- user accounts, setting, 83
- user names, including in e-mail, 81
- User Permissions command, 83
- User Policy command, 83
- user-created sites, enabling, 83
- User-Defined Workflows setting, 82
- Web Page Security Violation setting, 81
- web part connections, enabling, 83
- web part editing, enabling, 83
- Web Part Security command, 83
- web part updates, enabling, 83
- web services, returning, 180
- web.config file modifications, returning, 180
- Windows authentication, enabling, 78
- Workflow Task Notifications setting, 82

Web event receivers, 126

Web events

- absolute URL, 328
- list of, 327
- overview, 327
- parent web, GUID, 328
- server-relative URL, new server, 328
- server-relative URL, old server, 328
- SPWeb object, 328
- SPWebEventReceiver, 327–328
- triggering user, display name, 328
- triggering user, login name, 328
- web site ID, 328

Web events, SPWebEventProperties object

- FullUrl property, 328
- NewServerRelative Url property, 328

- Web events, SPWebEventProperties object, *continued*
 - ParentWebId property, 328
 - ServerRelative Url property, 328
 - UserDisplayName property, 328
 - UserLoginName property, 328
 - Web property, 328
 - WebId property, 328
- Web Page Security Violation setting, 81
- Web part configuration
 - AspNetHostingPermissions, 653
 - bug alert, 653
 - CAS (Code Access Security), 647–653
 - connection strings, 643–647
 - displaying data from databases, 643–647
 - permissions, 653
 - SecurityPermissions, 653
 - SharePointSecurityPermission, 653
 - SystemSecurityPermission, 653
 - trust levels, 650
- Web part connections
 - consumers, 637
 - dynamic, 637
 - overview, 636–637
 - providers, 637
 - sample project, 637–643
 - static, 637
- Web part hosts, 466
- Web Part Page content type, 466
- Web Part Page type. *See also* Web parts.
 - adding to pages, 48
 - creating, 47–48
 - defaults, 48
 - definition, 46
- Web part pages, 580
- Web Part project items, 124
- Web Part Security command, 83
- Web Part SharePoint element, conflict resolution, 776
- Web Part tab functions, 254–255
- Web Part Title function, 254
- Web parts. *See also* SharePoint pages; Web Part Page type.
 - adding, 48–51
 - adding, enabling, 93
 - ASP.NET 2.0, 580
 - closing, 255
 - configuring, 51–53
 - connections, adding, 255
 - connections, enabling, 83
 - connections, properties, 255
 - Content Editor, 50
 - definition, 580
 - editing, 255
 - editing, enabling, 83, 93
 - height, setting, 255
 - hiding, 255
 - HTML Form, 50
 - Image Viewer, 50–51
 - interconnections, enabling, 93
 - minimizing, 255
 - older SharePoint style, 580–581
 - online gallery access, enabling, 93
 - overview, 579–580
 - Page Viewer, 50
 - Picture Library Slideshow, 50
 - properties, 254
 - Relevant Documents, 50
 - saving to files, 255
 - saving to site galleries, 255
 - security, configuring, 93
 - Silverlight Web Part, 50
 - Site Users, 50
 - titles, setting, 254
 - types of, 580
 - updates, enabling, 83
 - User Tasks, 50
 - visual. *See* Visual web parts.
 - width, setting, 255
 - XML Viewer, 50
 - zone, changing, 255
- Web parts, code-only (sandboxed)
 - CreateChildControls() method, 612
 - creating, 610–619
 - overview, 609
 - SharePoint online, 609
- Web parts, Silverlight
 - AnimationSequence property, 627
 - breakpoints, 634–636
 - creating, 620–630
 - debugging, 634–636
 - DependentValuePath property, 627
 - deploying, 630–636
 - IndependentValuePath property, 627
 - installing Silverlight, 620
 - IsSelectionEnabled property, 627
 - ItemBinding property, 627
 - overview, 619–621
 - pie chart, creating, 620–630
 - PieSeries control, 623–627
 - testing, 634–636
- Web Parts setting, 68
- Web Parts SharePoint element, sandboxed solutions, 778
- Web property
 - event receivers, 320
 - SPContext object, 172
 - SPFile object, 213

- SPListItem object, 201
- SPWebEventProperties object, 328
- Web scope, features, 700–701
- Web services
 - calling, 532
 - returning, 180
 - workflows as, 532
- Web-based list views, 16
- WebBrowsable attribute, 603
- web.config file, 180
- WebConfigModifications property, 180
- WebDescription attribute, 603
- WebDisplayName attribute, 603
- WebId property
 - SPListEventProperties object, 322
 - SPWebEventProperties object, 328
- Webs property
 - SPSiteDataQuery object, 224
 - SPWeb object, 188
- WebService property, 180
- WebServiceFault workflow activity, 532
- WebServiceInput workflow activity, 532
- WebServiceOutput workflow activity, 532
- WebTemplate property, 189
- WebUrl property
 - SPItemEventProperties object, 325
 - SPListEventProperties object, 322
 - SPWorkflowEventProperties object, 330
- What's New Notification content type, 468
- While loops, 531
- While workflow activity, 531
- Width function, 255
- Wiki Page content type, 466
- Wiki pages, 46–47, 466
- Wiki pages, libraries of, 34
- Wildcard filter, 374
- Windows authentication, enabling, 78
- Word documents. *See* Content types, Word documents.
- Work items, creating, 469
- Workflow, SharePoint element, 707, 709
- Workflow activities
 - categories of, 529
 - compensation-related, 531
 - connection to WCF end-points, 532
 - container, 529
 - control flow, 529
 - decision points, 529
 - e-mail, sending, 534
 - exception handling, 530
 - exceptions, raising, 532
 - executing .NET code, 530
 - executing tasks, 529
 - faults, sending to the web service, 532
 - getting data from an external host, 530
 - grouping, 529
 - If/Else constructs, 530
 - If-Then-Else expressions, storing, 532
 - item-related, 532
 - iterating through elements, 531
 - listening to external events, 530
 - overview, 528–529
 - parallel execution, 531
 - RuleSets, storing, 532
 - sending data to an external host, 530
 - sequential execution, 531
 - standard, 529
 - state machine state, setting, 534
 - task-related, 532
 - transaction scope, 531–532
 - web services, calling, 532
 - While loops, 531
- Workflow activities, general
 - CallExternalMethod, 530
 - Code, 530
 - ConditionedActivityGroup, 530
 - Delay, 530
 - EventDriven, 530
 - EventHandlingScope, 530
 - FaultHandler, 530
 - HandleExternalEvent, 530
 - IfElse, 530
 - Listen, 530
 - Parallel, 531
 - Replicator, 531
 - Sequence, 531
 - Terminate, 531
 - While, 531
- Workflow activities, not supported by SharePoint
 - CompensatableSequence, 531
 - CompensatableTransaction, 531
 - Compensate, 531
 - InvokeWebService, 532
 - InvokeWorkflow, 532
 - Policy, 532
 - ReceiveActivity, 532
 - SendActivity, 532
 - Suspend, 532
 - SynchronizationScope, 532
 - Throw, 532
 - TransactionScope, 532
 - WebServiceFault, 532
 - WebServiceInput, 532
 - WebServiceOutput, 532
- Workflow activities, SharePoint-specific
 - ApplyActivation, 533
 - CheckInItemActivity, 534

- Workflow activities, SharePoint-specific, *continued*
 - CheckOutItemActivity, 534
 - CompleteTask, 533
 - CopyItemActivity, 535
 - CreateItemActivity, 535
 - CreateTask, 533
 - CreateTaskWithContentType, 533
 - DeleteItemActivity, 535
 - DeleteTask, 533
 - EnableWorkflowModification, 533
 - InitializeWorkflow, 533
 - item-related, 532
 - LogToHistoryListActivity, 533
 - OnTaskChanged, 533
 - OnTaskCreated, 533
 - OnTaskDeleted, 533
 - OnWorkflowActivated, 534
 - OnWorkflowItemChanged, 534
 - OnWorkflowItemDeleted, 534
 - OnWorkflowModified, 534
 - RollbackTask, 534
 - SendEmail, 534
 - SetState, 534
 - task-related, 532
 - UpdateAllTasks, 534
 - UpdateItemActivity, 535
 - UpdateTask, 534
 - workflow-related, 532
- Workflow activities, tasks
 - change events, 533
 - creating, 533
 - creating with content type, 533
 - creation events, 533
 - deleting, 533
 - deletion events, 533
 - marking complete, 533
 - rolling back, 534
 - updating, 534
- Workflow Association Form project items, 125
- Workflow Associations subnode, 101–102
- Workflow Initiation Form project items, 125
- Workflow Settings, 69
- Workflow Task Notifications setting, 82
- Workflow Templates subnode, 101–102
- Workflow1.cs file, 542
- Workflow1.designer.cs file, 542
- WorkflowAssociation property, 197
- Workflow-related workflow activity, 532
- Workflows
 - activation events, 534
 - associating with content types, 521–522
 - associations, displaying, 101–102
 - customizing, 30
 - data-driven workflows, creating, 530
 - delaying, 530
 - event-driven, grouping, 530
 - events. *See* List workflow events.
 - forms, 552–562
 - initializing, 533
 - invoking, 532
 - logging history list, 533
 - modification, enabling, 533
 - modification events, 534
 - overview, 525–528
 - sequential, 526
 - state machine, 527–528
 - status, updating, 533
 - suspending execution, 532
 - synchronizing execution, 532
 - templates, displaying, 101–102
 - terminating, 531
 - as WCF services, 532
 - as web services, 532
- Workflows, files
 - copying, 535
 - creating, 535
- Workflows, in SharePoint Designer
 - debugging, 562
 - declarative workflows, 562
 - importing to Visual Studio, 570–578
 - publishing, 569
 - reusable workflow, creating, 563–567
 - sequential workflows, 562
 - tasks, creating, 568–569
 - testing, 569–570
 - vs.* Visual Studio, 562
- Workflows, items
 - change events, 534
 - checking in/out, 534
 - copying, 535
 - creating, 535
 - deleting, 535
 - deletion events, 534
 - updating, 535
- Workflows, SharePoint
 - associating with lists, 536–540
 - breakpoints, 546–547
 - content type associations, 548–552
 - correlation tokens, 544
 - deploying, 546–548
 - Elements.xml file, 542
 - event handler, setting, 545
 - Features folder, 541–542
 - files, 542–543
 - list associations, 548–552



- multi-list, 552
- Packages folder, 541–542
- SharePointProjectItem.spdata file, 543
- site associations, 548–552
- tasks, creating, 543–545
- tasks, referencing, 546
- from Visual Studio 2008, 541–542
- Workflow1.cs file, 542
- Workflow1.designer.cs file, 542

Workflows property

- SPListItem object, 202
- SPWeb object, 188

Workflows setting, 69

Workspaces. *See* Subsites (workspaces).

X

XML elements. *See also* SharePoint elements.

- ActivationDependency, 703
- AddContentFieldType, 705
- ApplyElementManifests, 705
- code sample, 702
- CustomUpgradeAction, 705
- descriptions, 703–704
- ElementFile, 703

- ElementManifest, 701–706
- ElementManifests, 701–705
- MapFile, 705
- Parameter, 705
- pointing to, 704
- Property, 704
- UpgradeActions, 704–706
- VersionRange, 705

Xml property

- SPListItem object, 201
- SPListItemCollection object, 207
- SPUser object, 226

XML representation of user data, 226

XML Viewer web parts, 50

XMLDATA format

- converting folders, 207
- converting list items to, 201, 207

XSLT, custom list views

- customizing, 254
- reverting to default, 254

Z

- Zone Change function, 255
- Zone property, 325