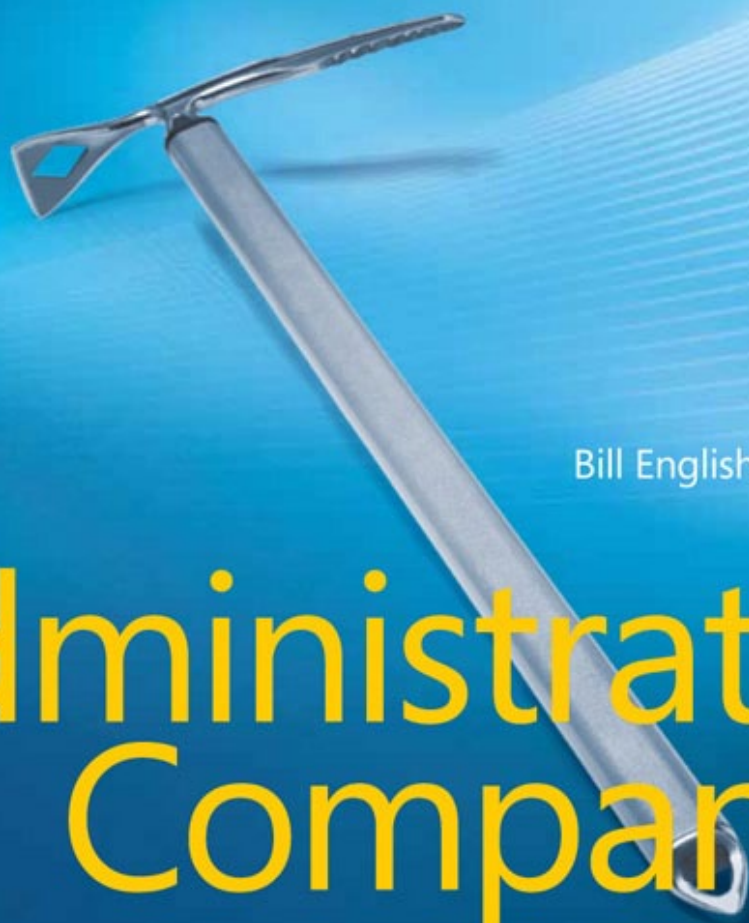


Microsoft®

Microsoft®

SharePoint® 2010

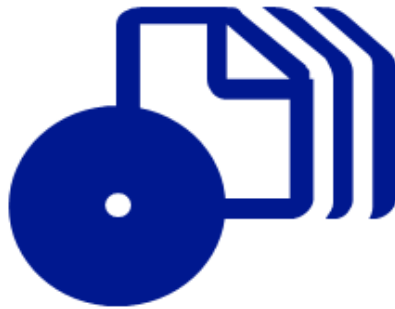


Bill English, Brian Alderman,
Mark Ferraz

Administrator's Companion



How to access your CD files



The print edition of this book includes a CD. To access the CD files, go to <http://aka.ms/627208/files>, and look for the Downloads tab.

Note: Use a desktop web browser, as files may not be accessible from all ereader devices.

Questions? Please contact: mspinput@microsoft.com

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2011 by Bill English

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2010933775

ISBN: 978-0-7356-2720-8

Printed and bound in the United States of America.

3 4 5 6 7 8 9 10 11 QGT 7 6 5 4 3 2

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to msspinput@microsoft.com.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Martin DelRe

Developmental Editor: Karen Szall

Project Editor: Karen Szall

Editorial Production: Custom Editorial Productions, Inc.

Technical Reviewer: Mitch Tulloch; Technical Review services provided by Content Master, a member of CM Group, Ltd.

Cover: Tom Draper Design

Body Part No. X17-09647

[2012-03-16]

To the entire SharePoint Community—a group of outstanding, talented people from whom I've learned more than I ever thought possible.

—BILL ENGLISH

To my dad, Donald K. Alderman, who always inspired me with his motivation, dedication, and commitment to everything he did and everyone he met.

—BRIAN ALDERMAN

To my lovely wife, Megan, and my dearest daughter, Piper. You mean everything to me.

—MARK FERRAZ

Contents at a Glance

<i>Acknowledgments</i>	<i>xxvii</i>
<i>Introduction</i>	<i>xxxii</i>

PART I THE BASICS

CHAPTER 1	Understanding the Basics of Collaboration in SharePoint 2010	3
CHAPTER 2	Understanding the Architecture of SharePoint 2010	57

PART II INSTALLATION AND IMPLEMENTATION

CHAPTER 3	Optimizing SQL Server for a SharePoint 2010 Implementation	95
CHAPTER 4	Installing SharePoint 2010	135

PART III BUILDING AND MANAGING A SHAREPOINT FARM

CHAPTER 5	Using Windows PowerShell to Perform and Automate Farm Administrative Tasks	195
CHAPTER 6	Managing SharePoint 2010 Using Central Administration	243
CHAPTER 7	Scaling Out a SharePoint Farm	345
CHAPTER 8	Information Management Policies	397

PART IV ORGANIZING, PUBLISHING, AND FINDING INFORMATION

CHAPTER 9	Organizing Information	437
CHAPTER 10	Collaboration and Portals	473
CHAPTER 11	Search Server 2010 and FAST Search: Architecture and Administration	525
CHAPTER 12	Using Windows PowerShell to Manage Search Services and FAST Search	573
CHAPTER 13	Customizing the Search Results and Search Center	615
CHAPTER 14	Administering Enterprise Content Management	673
CHAPTER 15	Administering Web Content Management and Publishing	725
CHAPTER 16	Securing Information	779
CHAPTER 17	Data Protection, Recoverability, and Availability	825

PART V ADVANCED TOPICS

CHAPTER 18	Aggregating External Data Sources	881
CHAPTER 19	Web Parts and Their Functionality in SharePoint 2010	921
CHAPTER 20	Publishing SharePoint 2010 to Mobile Devices	953
CHAPTER 21	Business Intelligence, Reporting Services, and PerformancePoint Services	989
CHAPTER 22	Upgrading to SharePoint 2010	1035
CHAPTER 23	Creating and Managing Workflows	1077
	<i>Index</i>	1123

Contents

<i>Acknowledgments</i>	<i>xxvii</i>
<i>Introduction</i>	<i>xxxix</i>

PART I THE BASICS

Chapter 1 Understanding the Basics of Collaboration in SharePoint 2010	3
Introducing SharePoint 2010 Capabilities	5
Using SharePoint Sites and Templates	8
Interacting with SharePoint Sites	10
Site Creation in SharePoint 2010	10
Editing Features in SharePoint 2010	12
Rich Text Editing	13
The SharePoint Ribbon	13
SharePoint Dialog Boxes	14
Creating Rich Themes	14
Applying the Upgraded 2010 UI to All Sites	18
Incorporating Rich Media Support	18
Working with Lists and Libraries	20
List Enhancements	20
List Input Form	20
List Templates	25
Creating a List	26
Managing List Columns	27
Advanced List Concepts.	36

What do you think of this book? We want to hear from you!
Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Large List Support	36
Site Columns	39
Editing List Input Forms	41
Microsoft Office Integration	49
SharePoint Workspace 2010	49
Microsoft Word 2010	53
Support for Other Microsoft Office Products	54
Summary	55
Chapter 2 Understanding the Architecture of SharePoint 2010	57
Enterprise Architecture	58
Modular	58
Extensible	58
Scalable	59
Logical Architecture Components	59
Service Architecture	59
Operating System Services	62
Database Services	62
Workflow Services	62
Supporting Services	62
Core Services	65
Capabilities	75
Sites	76
Content	76
Communities	77
Search	77
Composites	78
Insights	78
Deployment	80
Server Roles	81
Deployment Topologies	82
Development and Testing Environments	86
Application Pools	89
Understanding Application Pools	89

Preparing for SharePoint 2010 Installation.	141
Hardware Requirements	142
Software Requirements	142
Active Directory	147
SharePoint 2010 Preparation Tool	149
Performing SharePoint 2010 Installations	155
SharePoint 2010 GUI Installations	155
Performing a Command-Line Installation of SharePoint	177
Configuring a SharePoint 2010 Installation	179
Running the Farm Configuration Wizard	180
Renaming the Central Administration Database	184
Uninstalling SharePoint 2010	188
Summary.	191

PART III BUILDING AND MANAGING A SHAREPOINT FARM

CHAPTER 5 Using Windows PowerShell to Perform and Automate Farm Administrative Tasks

195

Using Windows PowerShell: The Basics.	195
What's New in Windows PowerShell 2.0	196
Windows PowerShell Architecture	197
Installing Windows PowerShell	200
Managing Systems Remotely with WinRM	203
Working with the SharePoint 2010 Management Shell	210
Understanding cmdlets	214
Getting Help	222
cmdlets for SharePoint Foundation 2010	230
cmdlets for SharePoint 2010	231
Performing Basic Administrative Tasks	232
Examples of SharePoint Administrative Tasks	234
Deploying SharePoint 2010 with Windows PowerShell Scripts	234
Managing SharePoint Services	237
Using Windows PowerShell During the Upgrade Process	238
Summary.	241

CHAPTER 6 Managing SharePoint 2010 Using Central Administration	243
Introducing Central Administration	243
Navigating the Central Administration Home Page	246
Central Administration Site Actions Menu	247
Central Administration Breadcrumb Navigation Trail Icon	248
Central Administration Browse Option	249
Central Administration Page Option	250
Performing Administrative Tasks Using Central Administration	252
Application Management	253
System Settings	289
Monitoring	299
Backup and Restore	315
Security	325
Upgrade and Migration	331
General Application Settings	333
Configuration Wizards	343
Summary	343
CHAPTER 7 Scaling Out a SharePoint Farm	345
History of SharePoint Farms	346
Services Federation	347
Single and Cross-Farm Services	350
Services Applications Logical Architecture	350
Service Application Proxy Groups	352
Publishing Service Applications	353
Scaling Service Applications Architecture	354
Web Tier	354
Application Tier	354
Database Tier	355
Identifying a Logical Location of Services on Servers	356
Planning Service Applications Architecture	357
Planning Topology Architecture	361

Configuring Service Applications.	364
Starting and Stopping a Service	364
Deploying Service Applications to an Existing Web Application	365
Creating a Custom Application Proxy Group for a Web Application	366
Creating a New Instance of a Service Application	367
Modifying the Application Pool of a Deployed Service Application	369
Modifying the Service Applications in the Default Application Proxy Group	371
Publishing Service Applications	373
Publishing Service Applications to Remote Farms	376
Step 1: Exchange Trust Certificates with the Remote Farm	376
Step 2: Explicitly Publish the Service Application	384
Step 3: Explicitly Connect the Service Application	385
Topologies for SharePoint 2010	387
Server Roles	388
Scaling Out a Farm with Server Groups	388
Summary.	395

CHAPTER 8 Information Management Policies 397

Introducing Records Management and Information Management Policies.	397
What Is Records Management?	398
Compliance Requirements Document	399
Records Management Roles	399
The File Plan	400
What Are Information Management Policies?	402
Planning Document Policies	403
Policy Metadata	404
Implementing and Configuring Information Management Policies . . .	404
Defining a Retention Policy	406
Generating Information Management Policy Usage Reports	411
Viewing Information Management Usage Reports	413

Implementing and Configuring a Records Center	416
Creating and Managing a Content Type	416
Creating the Records Center	417
Creating Records Libraries	421
Defining Required Metadata	422
Applying Information Management Policies	422
Creating Content Organizer Rules	422
Creating a Farm Level Send To Option	423
Placing a Hold on Records	424
Generating a File Plan Report	426
Generating an Audit Report	428
Implementing and Managing In Place Records	429
Implementing In Place Records at the Site Collection	429
Configuring In Place Records in a List or Library	430
Summary	434

PART IV ORGANIZING, PUBLISHING, AND FINDING INFORMATION

CHAPTER 9 Organizing Information	437
Developing an Information Architecture with SharePoint 2010	437
Value of Information	438
What Is Putability?	439
What Is Findability?	440
SharePoint 2010, Putability, and Findability	446
Putability and the Managed Metadata Service	449
Building an Information Architecture	451
Business Taxonomy	454
Information Architecture, Governance, and Requirements	454
Putability and Operational Taxonomies	459
Usability and Tool Development	461
Use of SharePoint and Maintenance	462
An Information Organization Project	462
Phase 0: Information Organization Assessment	463

Phase 1: Business Requirements Development	465
Phase 2: Technical Requirements and Project Charter	466
Phase 3: Audit and Analysis	467
Phase 4: Development of Putability and Findability	469
Phase 5: Governance and Maintenance	470
SharePoint 2010 Tools to Organize Information	471
Summary	472

CHAPTER 10 Collaboration and Portals 473

Using Collaboration Sites.	474
Collaboration Site Templates	475
Meetings Workspace Site Templates	476
Enabling Collaboration Features	477
Planning for Collaboration	479
Integration with SharePoint Workspace 2010	484
Choosing to Use Portal Sites.	485
Portal Site Scenarios	486
Publishing Portal Features	489
The Benefits of Enterprise Wikis.	490
Sharing Knowledge: The Social Experience	491
Social Features	491
Integrating with the Outlook Social Connector	496
Creating an Information Repository with the User Profile Service.	497
Uses and Benefits of the User Profile Service	497
User Profile Service Architecture	499
Setting Up and Configuring the User Profile Service	501
Profile Synchronization	515
Setting Up My Sites	519
Summary	523

CHAPTER 11 Search Server 2010 and FAST Search: Architecture and Administration 525

Introducing Enterprise Search	526
Search Architecture	528

Search Tools	529
Search Components and Processes	532
Farm and Application Configuration	536
Search Support Staff	536
Farm-Wide Search Settings	536
Managing Crawler Impact Rules	537
Creating Search Applications	538
Configuring the Search Application	540
Configuring Crawls	541
Creating and Managing Content Sources	542
Creating and Managing Crawl Rules	544
Using Server Name Mappings	548
Controlling Host Distribution	548
Managing File Types	550
Resetting the Index	550
Managing Crawls	550
Global Crawl Management	551
Content Source Crawl Management	552
User Crawl Management	552
Scheduling Crawls	552
Reviewing and Troubleshooting Crawls	554
Using Crawl Logs	554
Using Crawl Reports	558
Diagnostic Logging	559
Managing the Search Service Topology	561
Scaling Considerations	562
Modifying the Topology	563
Crawl Databases	564
Crawl Components	564
Property Databases	565
Index Partitions and Query Components	565
FAST Search Server 2010 for SharePoint	566
Introducing FAST Search Server 2010	566
Architecture and Topology	568
Summary	572

CHAPTER 12 Using Windows PowerShell to Manage Search Services and FAST Search 573

Working with Basic Search	574
Using Enterprise Search	577
Creating and Managing Search Application Topology	578
Administering Content Sources	587
Modifying Authoritative Pages and Search Metadata	594
Deploying and Managing FAST Search with Windows PowerShell	601
Using the FAST Search Server 2010 for SharePoint Shell	604
cmdlets for FAST Search Server 2010 for SharePoint	606
Adding FAST Search Server 2010 for SharePoint to a SharePoint 2010 Installation	608
Summary.....	613

CHAPTER 13 Customizing the Search Results and Search Center 615

Benefits of the Search Features	615
Introducing SharePoint Search	616
Improving the User Experience	619
Contextual Promotions	620
Customizing the User Interface	631
Creating New Search Pages	632
Customizing Search Pages.....	635
Query Pages	635
Search Box Web Part	636
Advanced Search Pages	640
Customizing the Results Presentation	644
Customizing Search Results Metadata	654
Improving Search Quality	656
Linguistics	656
Creating Search Rules	661
Rank Profile	663
XRANK	667
Search Quality Summary	669

Improving Collection Quality	669
Preventing Content Errors	669
Diagnosing Content Issues	670
Summary.	671

CHAPTER 14 Administering Enterprise Content Management 673

Document Management	674
Document Libraries	674
Versioning	678
Information Management Policy	680
Opening and Saving Documents in a Document Library	682
Metadata Navigation and Filtering	683
The Content Organizer	685
Document Sets	689
Location-Based Metadata Default Values	691
Document Center Template	692
Enterprise Metadata Management	692
Content Types and Columns	694
Managed Metadata Fields	697
Term Store and Term Sets	698
Content Type Syndication	703
Records Management	709
Records Management and Upfront Preparation	709
Records Management Features	710
Improved Records Center Site	711
In-Place Records Management	713
eDiscovery and Hold	716
Retention	717
Scalability	719
Web Content Management.	720
Accessibility and Markup	720
Content Query	720
Ratings	720
Folders	721

Page Authoring	721
Themes and Branding	721
Digital Asset Management	721
New Content Types	722
Video Streaming Infrastructure	722
User Interface	723
Summary.....	724

CHAPTER 15 Administering Web Content Management and Publishing 725

Publishing Site Types and Features	726
Choosing the Right Site Type	727
Enabling Publishing Features	728
Publishing Workflows	729
Branding	732
Using Variations	733
SharePoint 2010 Platform Enhancements	734
Administering Publishing Sites	735
Security and Permissions	735
Site Structure and Navigation	736
The SharePoint Page Model	739
SharePoint Designer Settings	740
Control and Insight from Analytics	742
Authoring and Publishing Web Content.....	745
Web-based Content Authoring	745
Smart Client Content Authoring	746
Web Parts in Publishing Pages	748
Digital Media and Asset Libraries	750
Reusable Content	751
Configuring Content Caching.....	752
Content Caching Overview	753
BLOB Cache	753
Page Output Cache	757
Object Cache	764

Managing Content Deployment	767
Content Deployment Concepts	767
Content Deployment Considerations	769
Configuring Content Deployment	771
Summary.	778

CHAPTER 16 Securing Information 779

Introducing SharePoint Security.	779
Securing a SharePoint Farm	782
Farm Administrators Group	782
Service Account Configurations	783
Approving or Rejecting Distribution Groups	783
Configuring Information Rights Management	784
Configuring Information Management Policies	784
Securing a Web Application	786
Zones	787
Authentication Providers	787
How Zones, Web Applications, and Security Work Together to Provide Secure Solutions	788
Changing Authentication Providers for a Web Application	795
Managing Web Part Security	796
Self-Service Site Creation	797
Securing Site Collections	800
Custom Site Collection Policies	800
Auditing Activities in a Site Collection	800
Security Trimming for Navigation	801
Site Collection Administrators	801
Securing Sites.	801
Indexing Site and List Content	802
Site Permissions and Permission Inheritance	803
Securing Lists.	819
Content Approval	819
Versioning Settings	821
Draft Item Security	822

Windows PowerShell Security	822
Service Application Permissions	823
Summary	823

CHAPTER 17 Data Protection, Recoverability, and Availability 825

Introducing Disaster Recovery	826
The Importance of Redundancy	827
The Role of Backups	827
Restores	829
Disaster Recovery Planning	830
Education	831
Documentation	834
Preparation	837
Using SharePoint 2010 Disaster Recovery Tools	838
Versioning	838
The Two-Stage Recycle Bin	839
Central Administration	841
Windows PowerShell	843
STSADM	844
SQL Server Backups	845
Read-Only Content Databases	846
Unattached Content Databases	848
Performing Backups and Restores	849
Using Central Administration	850
Using Windows PowerShell	851
Using STSADM	861
Example of Performing a SharePoint 2010 Farm Backup and Restore	866
High-Availability Options in SharePoint 2010	871
SQL Database Mirroring	871
SQL Server Transaction Log Shipping	872
Combining Database Mirroring and Log Shipping	874
SQL Snapshots	875

SQL Server Failover Clustering	875
Ensuring a Full Farm Recovery	876
Summary.	878

PART V ADVANCED TOPICS

CHAPTER 18 Aggregating External Data Sources	881
What Are the Business Connectivity Services?	882
Using the Business Data Connectivity Service Application and Model.	883
Business Data Connectivity Layer	885
External Content Types	886
BDC Models and Resource Files	887
Understanding the BCS Security Options.	890
Authentication Modes	891
Authorization	894
Managing Data Connections	896
Setting BCS Permissions	897
Configuring Profile Page Creation	898
Creating BDC Models	900
Importing BDC Models	903
Creating a Profile Page	905
Creating External Data Actions	907
Modifying BDC Objects	909
Using External System Throttling	910
Building Composite Solutions.	912
External Lists	912
Business Data Web Parts	914
External Data Columns	914
External Data Search	916
External Data and User Profiles	918
Summary.	920

CHAPTER 19 Web Parts and Their Functionality in SharePoint 2010	921
Introducing Web Parts	921
Before You Begin Developing Web Parts	921
Managing Web Parts	922
Adding Web Parts	923
Editing Web Parts	923
Connecting Web Parts	927
Exploring New and Improved Out-of-the-Box Web Parts	927
Lists And Libraries Web Parts	928
Business Data Web Parts	930
Content Rollup Web Parts	933
Document Sets Web Parts	935
Filter Web Parts	936
Forms Web Parts	937
Media And Content Web Parts	938
Outlook Web App Web Parts	939
Search Web Parts	940
Social Collaboration Web Parts	943
Other Web Parts	945
The Data View Web Part	945
PerformancePoint Services in SharePoint 2010 Web Parts	947
Web Analytics	948
Office Client Web Parts	949
Silverlight Web Part	950
Summary	951
 CHAPTER 20 Publishing SharePoint 2010 to Mobile Devices	 953
Introducing SharePoint Mobile in 2010	954
Microsoft Office Mobile 2010	954
SharePoint Workspace Mobile 2010	955
Microsoft Office Web Applications	957
Mobile Search	966

Mobile Views	967
Previewing Mobile Access Device Options.....	969
Windows Phone	969
Other Devices	970
Setting Up SMS Alerts.....	970
Mobile Account Prerequisites	971
Configuring a Mobile Account at the Farm Level	971
Configuring a Mobile Account at the Web Application Level	972
Configuring a Mobile Account Using Windows PowerShell	973
Exploring Mobile Access Scenarios.....	973
Anonymous Mobile Browser Access	973
Authenticated Cross-Firewall Access	974
Authenticated Secure Gateway Access	974
Examining Common Firewall Configurations.....	974
Edge Firewall	975
Multi-homed Firewall	976
Back-to-Back Firewall	976
Secure Access Gateway Configuration	977
Setting Up UAG for SharePoint.....	978
Important Considerations	979
Setting Up SharePoint for Cross-Firewall Access	979
Creating Application Portal Trunk in UAG	981
Publishing SharePoint Through a Portal Trunk in UAG	984
Summary.....	988

CHAPTER 21 Business Intelligence, Reporting Services, and PerformancePoint Services 989

Decision Making and BI.....	989
Establishing a Common Language.....	991
Applying KPIs to Goals	991
Creating Scorecards with KPIs	992
Understanding Dashboards	993
SQL Server Reporting Services 2008.....	994

What's New in SQL Server Reporting Services 2008	995
Understanding the Architecture of SQL Server Reporting Services 2008	998
Installing and Configuring SQL Server Reporting Services 2008	1007
Deploying and Managing Reports	1016
PerformancePoint Services 2010	1018
What's New in PerformancePoint Services 2010	1018
Understanding the Architecture of PerformancePoint Services 2010	1020
Installing and Configuring PerformancePoint Services 2010	1022
Upgrading PerformancePoint Server 2007	1029
Migrating Content to Another SharePoint 2010 Location	1032
Summary	1034

CHAPTER 22 Upgrading to SharePoint 2010 **1035**

Introduction to SharePoint 2010 Upgrades	1036
Philosophy	1036
How It Works	1037
SharePoint 2010 Upgrade Types	1038
SharePoint 2010 Upgrade Scenarios	1038
Special Cases	1040
SharePoint 2010 Upgrade Paths	1042
Preparing to Upgrade to SharePoint 2010	1044
Ten Best Practices for Upgrading to SharePoint 2010	1044
Backing Up and Testing Your SharePoint Farm	1046
Running the Pre-Upgrade Checker Tool	1047
SharePoint 2010 Upgrade Considerations	1051
Upgrade Process Considerations	1051
Upgrading the SharePoint User Interface	1051
Upgrading Features	1052
Five Steps to a Successful Upgrade to SharePoint 2010	1052
Performing a Database Attach Upgrade	1053
Preparing the New Farm Environment for the Upgrade	1053

Attaching Your Existing Content Databases to Your New SharePoint 2010 Farm	1056
Performing an In-Place Upgrade	1061
Installing the In-Place Upgrade	1062
Configuring the In-Place Upgrade	1066
Performing a Hybrid Upgrade Approach	1072
Upgrading from SharePoint Portal Server 2003	1073
Performing Post-Upgrade Configurations	1074
Shared Service Provider Upgrade	1074
Security Configurations	1075
Summary	1075

CHAPTER 23 Creating and Managing Workflows 1077

Understanding Human Workflows	1078
Understanding System Workflows	1078
Workflows in SharePoint 2010	1079
Workflow Stages and Forms	1079
SharePoint 2010 Workflow Types by Packaging and Deployment	1080
SharePoint 2010 Workflow Types by Associations	1080
SharePoint 2010 Workflow Improvements	1081
Predefined Workflows of SharePoint 2010	1083
Workflow Modeling and Development Tools	1085
Microsoft Visio 2010	1086
SharePoint Designer 2010	1089
Visual Studio 2010	1093
Workflow Modeling Tools Comparison	1097
Planning for Workflow Deployment	1098
Identify Roles Involved	1099
Security Considerations	1100
Information Disclosure	1102
Setting Up Workflow Configurations	1104
Web Application Configurations	1105
Preserving Workflow History	1106

Deploying Workflows.....	1107
Deploying Predefined Workflows	1107
Deploying Declarative Workflows	1108
Deploying Custom Workflows	1108
Activating a Workflow	1108
Adding a Workflow Association	1109
Monitoring Workflows.....	1114
Workflow Status	1114
Workflow Reports	1117
Troubleshooting Workflow Issues	1119
Detecting Workflow Errors	1119
Resolving Workflow Errors	1119
Summary.....	1121
 <i>Index</i>	 1123

What do you think of this book? We want to hear from you!
Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Acknowledgments

Bill English Although there are three names on the cover of this book, there are many more who put time, blood, sweat, and effort into making this book a reality. To be sure, this book would not be in your hands if it wasn't for Karen Szall, editor at Microsoft Press.

How does one give ample credit to an editor who has shown significant patience, understanding, and professionalism in the face of difficulties? If I could, I'd instruct Microsoft to give Karen Szall a significant raise in pay with six months off in Maui and ask God to give her sainthood in the church. I honestly think that on this project, Karen endured more difficulties than all of my other editors combined and did so with grace, elegance, professionalism, and patience. She is a fantastic editor and an equally great person. My only hope is that she'll consent to work with me at some point in the future. Thanks, Karen, for keeping this book project going when it was stalled and understanding when difficulties arose. The entire SharePoint community is indebted to you.

I would be remiss if I didn't thank Mark Ferraz and Brian Alderman for being coauthors on this book with me. Mark is one of those guys who "gets it" about SharePoint. He consults full time on SharePoint in a Fortune 50 company and has the scars and experience to write great content. He's not really a guy who spends a lot of time building his "brand" in the SharePoint community, but he's rock solid on this product and is someone who everyone who reads this book should get to know. He's an enjoyable guy who is going to go places, I'm convinced. I've learned a great deal from Mark about SharePoint. Mark, thanks for writing great chapters and for helping us all understand this product better.

Brian Alderman comes from an educational background and did a great job writing some long chapters—the 100-page chapter on Central Administration is his. I admire his persistence in writing and sticking with something until it is done. He brought solid SQL knowledge to this book (something we missed in the last Administrator's Companion), and he helped us all better understand the relationship between SQL and SharePoint. Like Mark, Brian also doesn't do a lot in the SharePoint community, but he is also rock solid on this product and is a great guy to get to know. If you have a chance to take a SharePoint class through Mindsharp with Brian Alderman, I would highly advise that you do it. You won't be disappointed.

I also want to thank Penny Coventry, a great friend from Hinckley, UK, for writing the PowerShell chapters. I consider Penny the matriarch of SharePoint. Like me, she's been around since the 2001 days. She's quiet, unassuming, pleasant, and

smart. Real smart. And her husband, Peter, is an Oracle developer. Yet, sitting at their dinner table, you'd never know it. There is no arrogance or pretense about them. Penny could easily be an MVP or an MCM, in my opinion. She knows this product very well from both the developer and administrator perspectives. Penny and I have written on several book projects now, and each one has been a huge privilege for me. Thanks, Penny!

I met Nikander and Margriet Bruggeman at a Microsoft meeting in Redmond a number of years ago. We have written several books together. Although I've not been able to get to know them as I would like, I know that they write great content, on time, in style, and with the book's elements highlighted. They are great authors who consult in the Netherlands and surrounding area in Europe. If you have a chance to work with them, please do so. You'll walk away a better person as a result.

Jason Gallicchio, Peter Abrue, and Roger Taylor worked together on the scaling out chapter when one of my original authors had to back out due to being over-committed. This was their first writing project, and while I think they found it challenging, I also think they used their real-world experience to help us understand how to scale out a farm. Peter and Roger built and maintain the entire Recovery.gov website in the Amazon cloud. They get four million unique hits each day at that site, and yet, running the entire farm in the cloud, they handle it with only four WFE servers. They know how to scale out a farm for a large deployment. Thanks, guys, for jumping in at the last minute and for getting a good chapter written for this book.

Daniel Webster is a lifelong friend of mine. I first met him back in the mid-90s at the old Valley Micro training center (it was sold in the late 90s). Daniel is a man of integrity, and it shows through in his insistence on getting the details right. Daniel was responsible for the search chapters, and he received some assistance from Suzanne George in pulling them together. I think they did the best that could have been done to get the chapters done within the page count that was handed to them. Frankly, Search and FAST could be the subject of an entire book as large as this one, so to condense such a large amount of information into two chapters is a real trick. I think they succeeded.

Suzanne George also took on the first chapter near the end of this project when another author backed out. Thanks, Suz, for taking on this chapter and getting it done so swiftly. I appreciate it. I also want to thank Brett Lonsdale for writing the Web Parts chapter. Brett, I know that writing isn't your first love, but I do want to thank you for doing it. I also want to thank Subramanian Sivaramakrishnan for writing the base content for the InfoPath chapter.

Mitch Tulloch did a great job in the technical editor role for this book. He added value by finding technical mistakes or not-well-stated concepts. Thanks, Mitch, for doing a great TE job. And Megan Smith-Creed provided a steady hand managing the many pieces while Julie Hotchkiss provided a clear and consistent copy edit.

Martin DelRe helped get this project rolling from a contract standpoint. I've worked with Martin on a number of projects, and I was very sad to see him move to another set of books that won't include SharePoint. I'll miss working with you, Martin. I wish you all the best in your future endeavors.

Others who added value to this book (though a couple may not know it because we researched topics and learned from their posts) include Ben Curry, Jie Li, John Holliday, Amanda Perran, Apran Shah, Dan Evers, Dan Herzog, Daniel Galant, Daniel Kogan, Zach Rosenfield, Kirk Stark, Dan Wesley, Luca Bandinelli, Frank Morales, Ram Gopinatham, Rick Taylor, and Todd Bleeker.

Back here in Minnesota, I'd like to thank my friends for sustaining me during some highly difficult times over the last 12 to 18 months: Dave and Meryl McGauvran, Mark Schneider, Jay and Dawn Herman, Bill and Ann Kinney, and Howard and Julie Tripp. I also want to thank my lovely wife, Kathy, for her continual support and my two children, David and Anna, for coming together to create a great family in which to live.

Most importantly, I want to thank my Lord and Savior, Jesus Christ, for your love and faithfulness. You have given me the talent and opportunity to write, and without you, I would be lost forever.

Brian Alderman There are many people who supported me throughout this project that I want to thank. I want to begin by thanking Bill English for coauthoring the book, as well coordinating and managing the project with Microsoft Press. I also want to thank Mark Ferrarz for his contribution in authoring the book with us. I want to thank everyone at Microsoft Press that assisted in the project, including Karen Szall, Mitch Tulloch, Julie Hotchkiss, and Megan Smith-Creed. Thanks to Ben Curry for providing direction and assistance while writing the book. Thanks to my family and friends for their kind and encouraging words throughout the project.

Mark Ferraz Involvement in a book project of this magnitude can be very rewarding, but also consuming at times. Accordingly, I'd like to thank the following people for their enduring support, friendship, and/or individual contributions. First I'd like to thank my wife, Megan, who was willing to put up with the late nights and weekend work associated with this project, shortly after the birth of our daughter, Piper. I could not have participated in this project without your willingness to take on the extra load that made my involvement possible. Coauthor

Bill English, without whom none of this would have been possible, thank you for your enduring patience and support. You are a mentor to me, and I am honored to have been able to work on this project with you. Coauthor Brian Alderman, thank you for working to ensure we closed any gaps in our collaborative work product. I'd like to thank all of the editors and technical reviewers from Microsoft Press. I'd also like to thank Karen Szall, our content development manager, for her diligence, brilliant coordinative capability, and for keeping us all in line and on time. I'd also like to thank Subramanian Sivaramakrishnan, Ankush Bhatia, Jeremy Liner, and Tejaswi Redkar for their content contributions and for being such a pleasure to work with. I respect and admire the skill and knowledge with which you perform your craft and feel fortunate to have been able to work with such knowledgeable experts in developing the content for this book. Lastly, I'd like to thank the SharePoint Teams and Microsoft for creating such a wonderful edition of the product with SharePoint 2010. You are all an amazing group of people doing fantastic work. Keep it coming.

Introduction

Welcome to *Microsoft SharePoint 2010 Administrator's Companion*! If you're reading this introduction, chances are good that you're interested in SharePoint 2010 administration. As you might suspect, this book is intended to be a reference on how to best deploy and manage this exciting product.

But despite the title, we have not assumed that only administrators will pick up this book and use it. You might be a project manager, an information management specialist, a network or SharePoint architect, or a power user who wants to learn more about SharePoint 2010 administration, design, and best practices. We have written this book with a wide variety of interested readers in mind:

- **Architects** will find good information in these pages about how to design a SharePoint implementation.
- **Power users** will benefit greatly from reading about site administration, the site templates that are available, and the Web Parts that ship with this product.
- **Content creators** will learn how to use document libraries to their full advantage.
- **Project managers** will find this a handy reference when working with SharePoint-oriented projects.
- **Information management specialists** will find ideas about implementation best practices when building taxonomies using content types.
- **Compliance specialists** can learn how SharePoint 2010 works with record and document management with a view to meeting compliance requirements.

As you can see, there is solid information in this book for a wide variety of professionals who will interface with SharePoint 2010.

How to Use This Book

This book contains not only great information about designing, deploying, and managing a SharePoint 2010 implementation; we've also included elements to help you understand the concepts and ideas better.

Look for book elements such as the following:



Real World sidebar Everyone can benefit from the experiences of others. Real World sidebars contain elaboration on a theme or background based on the experiences of others who used this product during the beta testing period.

Note Notes include tips, alternative ways to perform a task, or additional information that needs to be highlighted.

Important Boxes marked Important shouldn't be skipped. (That's why they're called Important.) Here you'll find security notes, cautions, and warnings to keep you and your network out of trouble.

More Info Often there are excellent sources for additional information on key topics. We'll use these boxes to point you to recommended resources.



Security Alert Nothing is more important than security when it comes to a computer network. Security elements should be carefully noted and acted on.

Best Practices Advice for best practices that we have gained through our own technical experience is presented in these elements..



On the Companion Media In some cases, we've been able to provide related material on the book's companion CD. These pointers let you know what is available.

System Requirements

The following are the minimum system requirements to run the companion CD provided with this book:

- Windows XP, Windows Vista, or Windows 7 operating system with the latest service pack installed and the latest updates installed from Microsoft Update Service
- CD-ROM drive
- Internet connection
- Display monitor capable of 1024 x 768 resolution
- Microsoft Mouse or compatible pointing device
- Adobe Reader for viewing the eBook (Adobe Reader is available as a download at <http://www.adobe.com>)

About the Companion Media

The companion CD contains the fully searchable electronic version of this book and additional materials you might find useful. Full documentation of the contents and structure of the companion media can be found in the Readme.txt file on the CD.

Support

Every effort has been made to ensure the accuracy of this book and companion content. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article accessible via the Microsoft Help and Support site. Microsoft Press provides support for books, including instructions for finding Knowledge Base articles, at the following website:

<http://www.microsoft.com/learning/support/books/>

If you have questions regarding the book that are not answered by visiting the site above or viewing a Knowledge Base article, send them to Microsoft Press via e-mail to mspinput@microsoft.com.

Please note that Microsoft software product support is not offered through these addresses.

We Want to Hear from You

We welcome your feedback about this book. Please share your comments and ideas via the following short survey:

<http://www.microsoft.com/learning/booksurvey/>

Your participation will help Microsoft Press create books that better meet your needs and your standards.

NOTE We hope that you will give us detailed feedback via our survey. If you have questions about our publishing program, upcoming titles, or Microsoft Press in general, we encourage you to interact with us via Twitter at <http://twitter.com/MicrosoftPress>. For support issues, use only the e-mail address shown in the previous section.

Understanding the Basics of Collaboration in SharePoint 2010

- Introducing SharePoint 2010 Capabilities 5
- Using SharePoint Sites and Templates 8
- Editing Features in SharePoint 2010 12
- Working with Lists and Libraries 20
- Advanced List Concepts 36
- Microsoft Office Integration 49

In the past several years, a collaboration revolution has evolved. Now, traditional, single-purpose applications no longer satisfy users' hunger and drive to get more done. It was not so long ago that someone would write a document, send it by fax, e-mail, or postal service to colleagues, and then wait for their collaborative feedback to return. This process could take hours—or even days—before team members' changes could be incorporated in the original document to create a final version.

Today, managers in companies that are competing in the evolving global marketplace need to find ways to increase the efficiency of their workforce. Companies need their employees to be able to connect and work anywhere and anytime using a variety of devices to connect and complete their work.

SharePoint sites bring employees together to share information, data, and expertise by allowing interaction with others using standard Microsoft Office applications as well as standard Web browsers such as Internet Explorer or Firefox. The focus in this chapter is primarily on user collaboration through the Web browser and a general overview of collaboration in Microsoft SharePoint 2010.

SharePoint 2010 in Five Words

The SharePoint 2010 platform can be described in five essential words: collaboration, aggregation, organization, presentation, and publication. This brief overview, using those five words, describes what SharePoint is and what it does for your nontechnical users and decision makers.

- **Collaboration** The basis of the SharePoint platform is collaboration. Emanating from the original Microsoft SharePoint Team Services (STS) that was provided free on the Microsoft Office 2000 Professional CD, STS was renamed Windows SharePoint Services (WSS) and formed the foundation for the Microsoft SharePoint Portal Server 2003 (SPS) and the Microsoft Office SharePoint Server 2007 (MOSS) platforms. It is now called Microsoft SharePoint Foundation Services (SFS). The core of this foundation remains a robust collaboration platform that supports workflows, security trimming, asynchronous interaction between team members, and other features that create an environment for collaboration. SFS is free and requires neither Client Access Licenses (CAL), a server license or, if utilized on the Internet, an Internet connector license.
- **Aggregation** SharePoint 2010 allows you to aggregate information from a variety of dissimilar sources, such as databases, file servers, or websites. The aggregation technologies include, but are not limited to
 - Linking to the content
 - Indexing the content
 - Hosting the content
 - Really Simple Syndication (RSS) (if the content is published using RSS)
 - Business Connectivity Services (BCS) (if the content is in a database)The ability to aggregate content without necessarily hosting it is one of the strengths of SharePoint. A robust information management system must be able to aggregate information, especially if the aggregation of selected information is needed to combine that information into a collaborative effort.
- **Organization** In nearly all scenarios, when data is aggregated, it also must be organized. Nearly all of us who have been in the information technology (IT) field for any length of time can think of file servers, databases, My Documents, and other repositories that have become little more than dumping grounds for all types of information. IT professionals (IT Pros) use phrases like “out of control,” or “a total mess,” or “nothing but rubbish” to describe repositories of information that lack organization. SharePoint 2010 provides a number of organization features, which include but are not limited to
 - Managed Metadata Service

- Content types
- Site columns
- Managed paths
- Summary Links Web Part
- Sites directory

Organizing information is an important task for any company that uses SharePoint. Simply moving information into SharePoint or indexing it won't give you the *findability* that you and your users will need. You must actively and consistently organize your information to make it useful.

- **Presentation** After you have aggregated and organized your information, you'll need to present it in a pleasant and meaningful way. SharePoint Designer can help with the presentation layer. But so can the ability to present Business Intelligence (BI) information in a dashboard. Project data can be presented in meaningful ways using the project Web Parts in SharePoint 2010. All of the Enterprise Content Management (ECM) features, such as Document Management (DM), Records Management (RM), and Web Content Management (WCM), can be thought of as tools to accomplish aggregation, organization, and presentation.
- **Publication** SharePoint makes it possible to publish information created and developed by a small group of team members for viewing by a wider audience. The Web Content Management is one of the most important tools available for publishing information, but you also have major/minor versioning capabilities in document libraries that allow documents to be published as they are developed.

Introducing SharePoint 2010 Capabilities

SharePoint 2010 builds on the value of SharePoint Server 2007 and provides a core team collaboration experience along with a strong, consistent, development platform that can be used to tailor the SharePoint experience to meet specific corporate and user needs. Microsoft SharePoint 2010 is focused on providing and satisfying business needs by

- Connecting and empowering people
- Cutting costs with a unified infrastructure
- Rapidly responding to business needs

The enhanced features provided in this release of SharePoint can be categorized into groups that help describe the various editions and features available. Table 1-1 describes these capabilities in detail.

TABLE 1-1 SharePoint 2010 Capabilities

CAPABILITY	DESCRIPTION
Sites	Engage employees, partners, and customers by utilizing Web-facing sites, workspaces to share information, data, and expertise. These sites can be located either inside or outside a firewall.
Communities	Utilize social networking concepts to connect people, engage employees, and streamline information organization.
Content	Provides quick, accurate, and secure access to information by driving compliance, reducing risk, and consolidating systems using content and role-driven rules during the creation, review, publication, and disposal of content.
Search	Find, explore, and connect to information across SharePoint lists, sites, external systems, and data sources such as file shares, websites, or line-of-business applications.
Insights	Enable users to improve business decisions by empowering decision makers, improving organizational effectiveness, and enabling efficiency by turning raw data into actionable conclusions that drive business results through the sharing of data-driven analysis.
Composites	Allow users to quickly create integrated, customized solutions that support power users and professional developers.

With the release of SharePoint 2010, as with previous releases, there are two basic platforms: SharePoint Foundation 2010 and SharePoint 2010. Table 1-2 describes the editions of SharePoint that are available.

TABLE 1-2 SharePoint 2010 Editions

EDITION	DESCRIPTION
INTRANET SCENARIOS	
Enterprise Client Access License (CAL)	Enables advanced scenarios for end users; provides full interoperability with external line-of-business applications, Web services, and Microsoft Office client applications.
Standard Client Access License (CAL)	Enables deployment of a business collaboration platform across all types of content, simplifying content management and business process across an organization.
INTERNET/EXTRANET	
SharePoint 2010 for Internet Sites, Enterprise	For organizations that want to create scalable customer-facing Internet websites or private secure extranet sites using the full Enterprise capabilities of SharePoint 2010.

EDITION	DESCRIPTION
SharePoint 2010 for Internet Sites, Standard	For small and mid-sized organizations that want to create public Internet sites or basic extranets using the Standard features of SharePoint 2010.
ENABLING TECHNOLOGIES	
SharePoint Designer 2010	A tool for advanced users and developers who want to create SharePoint solutions in an easy-to-use environment. No-code solutions can be created for many scenarios including collaborative sites, Web publishing, line-of-business data integration, business intelligence solutions, and more.
SharePoint Foundation 2010	For smaller organizations needing a solution for secure, Web-based collaboration to coordinate schedules, organize documents, and use team workspaces, blogs, wikis, and document libraries.

SharePoint Foundation Services provides a low-cost, entry-level solution for secure, Web-based collaboration. SharePoint Foundation 2010 allows collaboration among users to coordinate schedules, organize documents, and participate in discussion forums through team workspaces, blogs, or wikis. Because SharePoint Foundation is the underlying infrastructure for SharePoint Server, straightforward upgrade paths are possible as corporate needs require additional functionality and advanced content capabilities. Table 1-3 summarizes the capabilities for SharePoint Foundation 2010.

TABLE 1-3 SharePoint Foundation 2010 Capabilities

CAPABILITY	DESCRIPTION
Sites	Ribbon user interface, SharePoint Workspace, SharePoint Mobile, Office Client and Office Web Application Integration, and Standards Support
Communities	Blogs and Wikis
Content	Remote BLOB Storage, List Enhancements
Composites	Business Connectivity Services, External Lists, Workflow, SharePoint Designer, Visual Studio, API Enhancements, REST/ATOM/RSS support

For organizations that require scalable business platforms, SharePoint 2010 has been divided into two editions: Intranet and Internet/Extranet; each of these software editions has different fee structures based on the product purchased. Both editions share the capabilities with SharePoint Foundation and are listed in Table 1-3; they also have the additional capabilities listed in Table 1-4.

TABLE 1-4 SharePoint 2010 Capabilities

CAPABILITY	DESCRIPTION
Communities	Blogs and Wikis *Tagging, Tag Cloud, Ratings *Social Bookmarking, My Sites, Activity Feeds, Profiles and Expertise, and Organization Browser
Content	Remote BLOB Storage, List Enhancements *Enterprise Content Type, Metadata and Navigation, Document Sets, Multistage Disposition, Audio and Video Content Types
Insights	PerformancePoint Services, Excel Services, Chart Web Part, Video Services, Web Analytics, SQL Server Integration, Power Pivot
Search	Social Relevance, Phonetic Search, Navigation, FAST Integration, Enhanced Pipeline
Sites	Ribbon UI, SharePoint Workspace, SharePoint Mobile, Office Client and Office Web Application Integration, and Standards Support.
Composites	Business Connectivity Services, External Lists, Workflow, SharePoint Designer, Visual Studio, API Enhancements, REST/ATOM/RSS support.

**Supported in SharePoint 2010 Server only.*

This chapter explores the basic collaboration techniques that are available utilizing the Sites capabilities provided in SharePoint 2010.

Using SharePoint Sites and Templates

SharePoint 2010 comes with site templates to help you begin the journey into user collaboration. These templates provide a basic framework design for building SharePoint sites to meet an organization's needs and requirements. More often than not, these templates are used by organizations out of the box, with very few—if any—customizations. If you understand how to use the different templates that are available when designing and implementing a SharePoint installation, you can help ensure that both farm administrators and nontechnical users choose the correct site template to allow employees in your organization to collaborate effectively. SharePoint 2010 site templates can be categorized as follows.

- *Collaboration* sites allow users to quickly author and convey information (press releases, project information, events, announcements, and so on) through one or more

Web pages, blogs, or wikis. These sites contain document libraries, calendar items, tasks, and discussions.

- *Content* sites assist with the management of documents, using version and change controls.
- *Publishing* sites help SharePoint sites adhere to corporate workflow policies and procedures. For example, an extranet portal provides access to corporate content in a secure manner. Generally, extranet sites are exposed to the public Internet and require additional update processes and procedures.
- *Web database* sites are new in SharePoint 2010 and allow for advanced functionality and allow modifiable database templates that can be used or modified as necessary.
- *Search* site functionality has been revamped in SharePoint 2010, and it now provides a better overall user search experience, providing features such as phonetic name matching and query suggestions. These sites include pages for search results and advanced searches. The SharePoint Server 2007 search sites such as the Search Center have been deprecated.

SharePoint 2010 allows for the combination of one or more of these site types to provide the best possible user experience platform. Before identifying which primary site templates to use, you must first determine the primary purpose of the site. Table 1-5 identifies the templates available in SharePoint 2010 (items new to SharePoint 2010 are shown in *italics*).

TABLE 1-5 SharePoint 2010 Templates

SITE TYPE	SHAREPOINT 2010 SITE TEMPLATE
Collaboration	Document Workspace
	<i>Group Work Site</i>
	<i>Issue Tracking</i>
	Meeting Workspaces—Basic, Blank, Decision, Multipage, and Social Meeting
	Team
	Wiki
Content and data	Blog
	Document Center
	Document Workspace
	My Site Host
	<i>Records Center</i>
	<i>Visio Process Repository</i>
Publishing	<i>Enterprise Wiki</i>
	Publishing Portal

continued on the next page

SITE TYPE	SHAREPOINT 2010 SITE TEMPLATE
Web database	<i>Assets Web Database</i>
	<i>Charitable Contributions Web Database</i>
	<i>Contacts Web Database</i>
	<i>Issues Web Database</i>
	<i>Projects Web Database</i>
Search	<i>Basic Search Center</i>
	<i>FAST Search Center</i>
	<i>Enterprise Search Center</i>

The team collaboration site provides much of the functionality of the other sites.

Interacting with SharePoint Sites

The SharePoint 2010 user interface (UI) provides better user interaction and simplifies feature and content adjustments. The relocation of the Site Actions button to the upper left along with new action tabs simplify the ways a user can interact with the site. The new Ribbon bars are context- and security-sensitive, which means that they change depending on which user is logged in and what level of security role the user has.

The Ribbon displays possible actions that are available within the site. For example, a site administrator will see the Page tab, which allows modification to the page and the ability to add SharePoint Web parts, while a user who is not an administrator will not see this tab. Other features, such as the Quick Launch menu and site search text box function, remain similar in look to SharePoint Server 2007.

Site Creation in SharePoint 2010

Users designated as farm administrators within SharePoint 2010 can define and group sites, along with their respective subsites, using a common URL namespace. For example, a company could have a site URL of *http://contoso* and create multiple subsites URLs such as *http://contoso/operations/engineering*, *http://contoso/operations/IT*, or *http://contoso/operations/facilities*. No physical limit exists to the URL depth, and URLs can be adjusted to suit the needs of the users who use that site.

A top-level site, such as *http://contoso*, with its associated subsites, is referred to as a *site collection*. There are additional administrative settings that will apply to the entire collection of sites rather than any one individual site within the collection. For example, site quotas define the total amount of data that can be hosted within a site collection. Site quotas are applied at the site collection layer and cannot be broken down into quotas for each individual site. By contrast, user security applies specifically to an individual site, an individual Web Part, or a specific list item. Hence, different configurations apply at different levels within the

SharePoint platform, and your users will need to learn which settings apply at which levels in order to administer their site collections and sites properly.

When a top-level site is created, subsites are added to target collaboration flow and functionality. For example, consider a company intranet website where management wants to control access to change content. To do this, the top-level site might be configured as a publishing site to limit updates and require user workflows for publishing. However, the company's engineering department requires an intranet subsite for updating documentation, management of wikis, and project management functions. A new subsite might be created for this purpose so that access can be granted, as required, by the engineering department but still keeping those functions quarantined within a site. For more information, read the Real World sidebar, "Site Collection Creation with Multiple Databases," in Chapter 6, "Managing SharePoint 2010 Using Central Administration."

SharePoint 2010 enhances the administration-user interface by employing Silverlight to pop up new pages as needed, thereby retaining the users' context within the site. By clicking Site Settings in the upper-left corner of the SharePoint 2010 Central Administration application window, the Site Actions menu appears. New quick links as well as the traditional site settings provide easy access to most administrative functions, as illustrated in Figure 1-1.

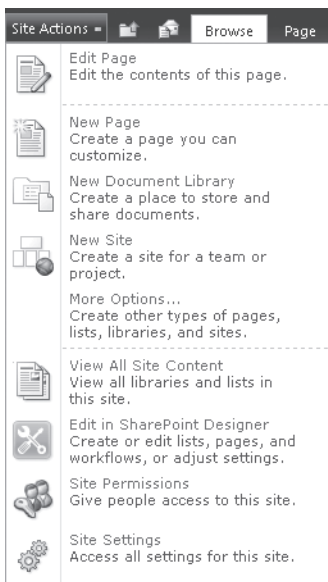


FIGURE 1-1 Site Actions menu

NOTE If you click the Site Actions button and the Site Actions menu does not appear, try adding the URL of the root site in the site collection to your trusted sites list in Internet Explorer.

The Site Settings link will take you to a page very similar to the SharePoint Server 2007 Site Settings menu. Although the page is formatted in a slightly different way, SharePoint 2010 navigation is similar enough to SharePoint Server 2007 to make navigation during the upgrade process easier for site and farm administrators. If Silverlight is installed, the new Silverlight administrative pages are accessible by clicking New Site in the Site Actions menu (refer back to Figure 1-1). This action opens the Create page shown in Figure 1-2. If Silverlight is not installed and you click New Site on the Site Actions menu, you will be taken directly to the New SharePoint Site page (Newsbweb.aspx).

From the Create page shown in Figure 1-2, choose the site template you want to use to create a new site. Then enter the site title and URL name and click the Create button. Because each site contains different lists and libraries, this chapter uses the Team Site template for demonstration purposes.

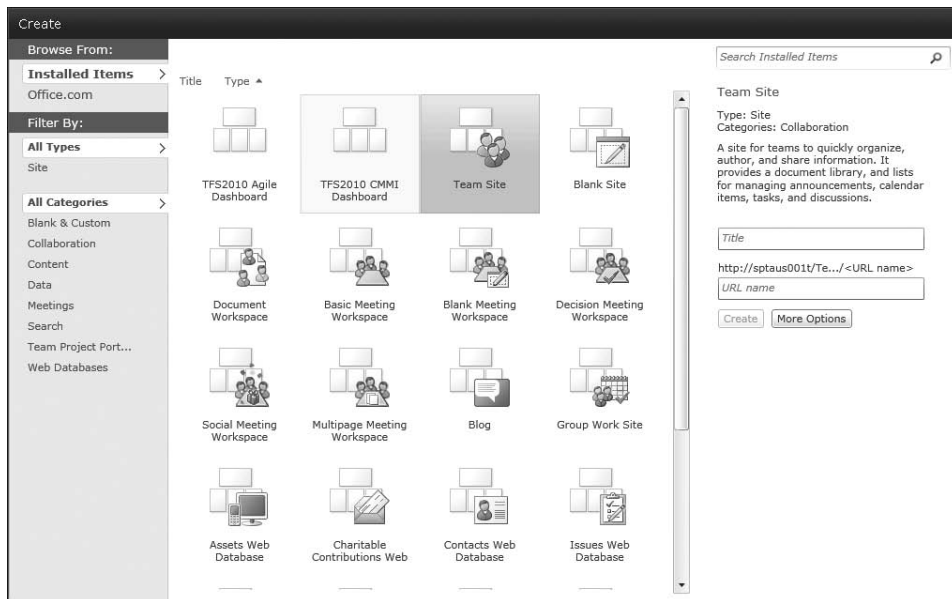


FIGURE 1-2 The Create page provides templates that you can select from to create a new site.

Editing Features in SharePoint 2010

SharePoint 2010 features a new and improved editing framework. With these enhancements, editing functionality becomes easier than it was in previous versions. One comment repeated over and over by power users when they start using SharePoint 2010 is "I didn't know SharePoint could do that!" SharePoint 2010 includes many common Microsoft Office functionalities, and as a result, users are more aware of the platform capabilities. Power users can modify and enhance the sites' content to provide collaboration and information to the user base. By default, the new Team Site home page has a Click To Edit feature, making content

changes simple. These new editing features encourage data and information collaboration among users that is easy to create, find, and use. In addition, these features will encourage you to make SharePoint your principal choice for collaboration.

Rich Text Editing

The new Team Site home page contains a Click To Edit feature that makes content changes simple. Power users can easily upload images, create links, and add Web Parts with a click of the mouse. Users with design or higher access can click the Page tab on the Ribbon or choose Edit Page from the Site Actions menu. In SharePoint 2010, the Edit Page command opens the Wiki Page Home Page feature; it is enabled or disabled as requirements dictate. If the Wiki Page is disabled, the Team Site will display as the traditional SharePoint Server 2007 home page. To check if this feature is enabled or disabled, complete the following steps.

1. Click Site Actions and then select Site Settings.
2. Under the Site Actions group, click Manage Site Features.
3. Click Active or Deactivate, as shown in Figure 1-3.



FIGURE 1-3 Wiki Page settings

The SharePoint Ribbon

SharePoint 2010 uses a Ribbon similar to the Microsoft Office 2007 and Office 2010 Ribbon and, just like Office, items are logically grouped. Further, the SharePoint 2010 Ribbon can be customized as needed and retains the users' security context. Figure 1-4 contains an example of the Ribbon for the Team Site home page.

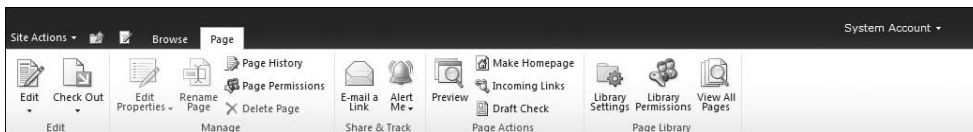


FIGURE 1-4 Team Site Edit Ribbon

As a page is edited, the Ribbon changes to make relevant operations available on the page or Web Part. For example, Figure 1-5 shows operations available for content editing on a page.

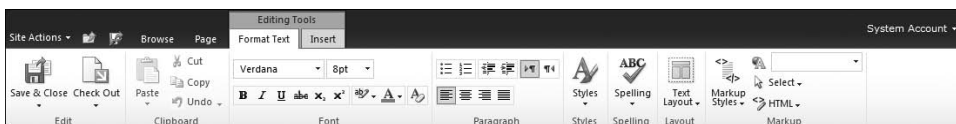


FIGURE 1-5 Team Site Edit Content Ribbon

SharePoint Dialog Boxes

New to SharePoint 2010 are the form dialog boxes, where functions such as creating a new item utilize forms that appear in the users' context window instead of redirecting the browser to a new page. To add a Web Part, users with proper permissions can click the Insert tab in the Action bar, select the Web Part button, choose the Web Part, and then click Add. During this process, the page context doesn't change. The ease with which power users can navigate and modify SharePoint sites should increase adoption and empower such users to do more with their sites. Further, the SharePoint 2010 wiki page allows users to insert Web Parts practically anywhere on the page.

Some of the most notable new Web Parts released with SharePoint 2010 are

- Silverlight viewer
- Media Web Part
- Chart Web Part
- HTML form Web Part
- iView Web Part
- Search Web Parts

One of the most frequently used Web Parts in SharePoint websites is the Content Editor Web Part. In SharePoint Server 2007, this Web Part allows users to enter information using a Microsoft Write experience. Using this Web Part, power users can type text inside the box and perform basic functions using—but not limited to—tables, fonts, and images. However, Microsoft Write is a very simple word-processing program, and administrators and users alike needed more features and capabilities than it could offer. Until SharePoint 2010 was introduced, however, the choices were either to teach HTML to the broad user base or have administrators or Web developers write code through the source editor.

SharePoint 2010 revamps and revitalizes the Content Editor Web Part so that it now resembles Microsoft Word more than Microsoft Write. No longer will users complain about broken URLs, tables not aligned when published, and so on. Business users will like this enhanced Web Part because it is so easy to use, styles can be applied to tags in Page Edit mode, and the user still has the ability to edit the HTML as necessary. Further, with the addition of the Ribbon, the Web Part easily converts to XHTML from the content editing Ribbon shown in Figure 1-5.

MORE INFO You can learn more about Web Parts and how they are managed, inserted onto a page, and removed from a page by reading Chapter 19, "Web Parts and Their Functionality in SharePoint 2010."

Creating Rich Themes

As companies strive to brand themselves in an ever-changing marketplace, websites must quickly adapt. SharePoint 2010 has revised and enhanced the way in which power users and administrators create and apply themes to sites. One of the most important changes to site

themes is the ability to inherit themes from the parent site. SharePoint 2010 makes theme changes significantly easier by providing three different methods for defining and changing themes.

- You can use Microsoft PowerPoint 2010, an application already known by most power users.
- You have access to enhanced Web-based site themes.
- You can create a CSS style sheet and attach to the master page.

Applying site themes assists content and Web managers in defining the content space by unifying the branding experience from other corporate media into the SharePoint website. Departments, divisions, and product lines can be defined using specific color palettes as needed.

Creating Themes Using the User Interface

You can create or change a basic color theme for a site directly through the UI for a site. Important feature enhancements to the Web interface include the ability to change the header and body font; the ability to change texts, accents, and hyperlinks; and a preview mode to view changes before they are applied to the site. To create or change a site, use the following steps.

1. Open the SharePoint site as a site administrator or Web designer.
2. Click Site Actions and then select Site Settings.
3. Under Look And Feel, choose Site Theme.
4. Click Select A Color to change a color of a specific element, or choose a theme from the list.
5. Click Preview to view the site changes before publishing.
6. Click Apply.

Creating Themes with Microsoft PowerPoint 2010

SharePoint 2010 no longer requires users to know and understand Cascading Style Sheets (CSS) to create a site theme. By utilizing a known application such as Microsoft PowerPoint 2010, power users can create and apply site themes quickly and easily. To change a website theme based on Microsoft PowerPoint 2010, follow these steps.

1. Open Microsoft PowerPoint 2010 and either create a new document or open an existing document. If you use an existing corporate PowerPoint document that already defines themes and colors for the enterprise, you might not need to perform steps 2 and 3.
2. Change the PowerPoint theme palette, if needed, by clicking the Design tab. Microsoft PowerPoint has many default themes available for you to use. Choosing a theme defines styles for the SharePoint site fonts, hyperlinks, and so on.

3. Choose a color palette for the site by clicking Colors. By default, there are many default color palettes available; choose one of the defaults or create a new color template. The colors chosen here define the site font colors for the SharePoint site.
4. Change the font styles as needed to further refine different text styles. Figure 1-6 shows a new Microsoft PowerPoint document with customized colors, fonts, and theme.

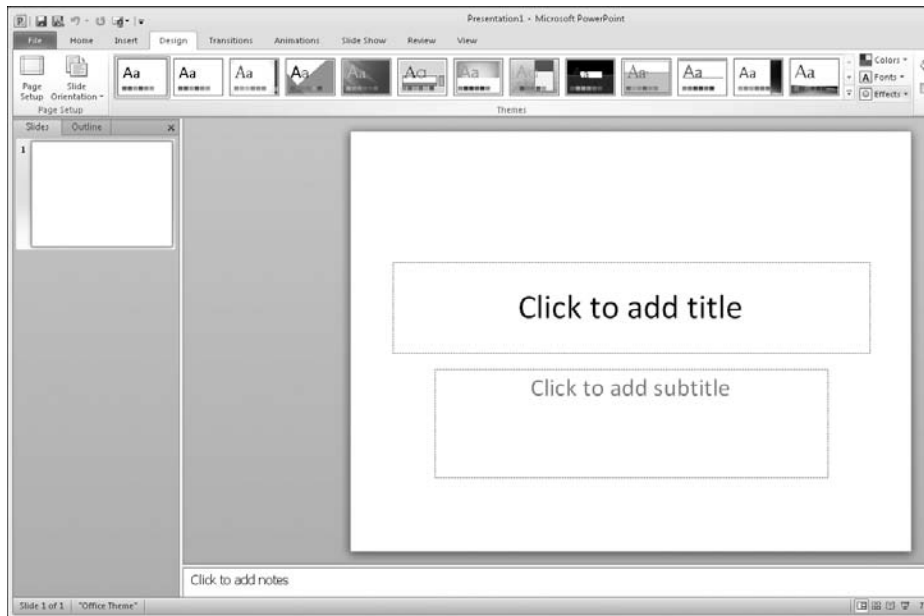


FIGURE 1-6 Changing a theme in PowerPoint 2010

5. Save the PowerPoint document as a Microsoft PowerPoint Theme (*.thmx).
6. Open your SharePoint 2010 top-level site, click Site Actions, and select Site Settings.
7. Under Galleries, click Themes.
8. Click Add New Item.
9. In the new window, browse to your file and keep / as the destination.
10. Click OK.
11. Open the SharePoint team site where you want to change the theme, click Site Actions, and then select Site Settings.
12. Under Look And Feel, choose Site Theme.
13. Select the newly uploaded theme and click OK.

After completing these steps, the SharePoint site will refresh to reflect the updated fonts, colors, and styles defined in the PowerPoint theme file. Additional customizations can be made, as necessary, by adjusting the theme file, via the user interface, or by creating customized style sheets.

Creating Themes Using Style Sheets

You may sometimes need to modify more than the colors and fonts of a website. You can create site themes using style sheets and then attach them to master pages through SharePoint Designer 2010. After you create a style sheet, you attach it to the master page by following these steps.

1. Copy your style sheet to the “14 hive” located by default at %ProgramFiles%\Common Files\Microsoft Shared\Web Server Extensions\14\Template\Layout\1033\Styles.

NOTE The exact location of the CSS style sheet depends on the language pack installed as the default. In the URL in this step, 1033 has defined English as the language Locale ID.

2. Open the website in SharePoint Designer 2010, as shown in Figure 1-7.
3. Open the master page for your site (the default for a site is default.master).
4. Click Attach Style Sheet from the Ribbon.
5. Choose whether to apply to all pages or just the current master page and then click OK.

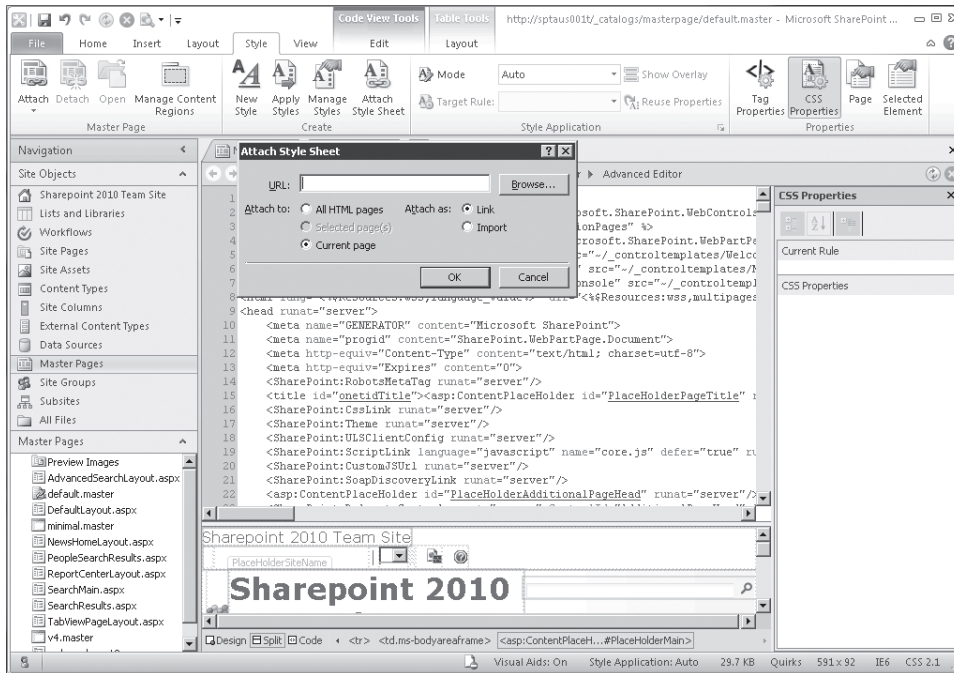


FIGURE 1-7 Changing a site theme in SharePoint 2010 Designer

MORE INFO Applying styles using CSS branding will not be described in depth in this chapter, but you can find more information on this topic at <http://blogs.msdn.com/sharepointdesigner/>.

Applying the Upgraded 2010 UI to All Sites

Because the UI has changed significantly in both SharePoint Foundation 2010 and SharePoint 2010, any customizations that rely on specific CSS classes and UI elements will work best for users in backward compatibility mode. When you initiate upgrades to either SharePoint Foundation 2010 or SharePoint 2010, the installer is able to choose either backward compatibility mode or the upgraded user interface mode. These modes may be toggled between backward compatibility and the new interface at the site collection level or site level.

- To preserve the look and feel of existing sites, choose the PSConfig or PSConfigUI.
- To preserve the look and feel of existing sites during an upgrade, you can attach the old content database to a new server farm using Stsadm.

You can use the Web interface to set all sites in a site collection to the upgraded UI (and prevent users from using the earlier interface). To do this, follow these steps.

1. Under Site Settings, click Site Collection Administration.
2. Click Visual Upgrade to display the page shown in Figure 1-8.

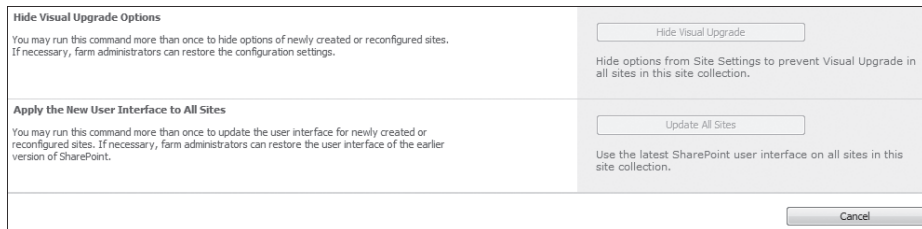


FIGURE 1-8 Apply Supported User Experience to all sites

3. Click Update All Sites to apply the new interface to all sites. Additionally, on this page, the site collection administrator can choose to prevent the upgrade of newly created sites in a site collection to the new experience. This is important if a SharePoint farm is unable to use the new look and feel for any reason.

Incorporating Rich Media Support

Rich media, a key component in the collaboration landscape, is another vehicle you can use to get targeted information to users. Training videos for employees, as opposed to on-site training sessions, are increasing in popularity as companies strive to reduce costs. Furthermore, in the “information now” age, users want information when they need it, where they need it, and within a few clicks of the mouse. A recent press release from Gartner, Inc., was titled “Gartner Predicts That by 2013, More Than 25 Percent of the Content That Workers

See in a Day Will Be Dominated by Pictures, Video or Audio" (at <http://www.gartner.com/it/page.jsp?id=834213>). Clearly there is a growing need for organizations to be able to store, find, and retrieve media in the same ways as documents. SharePoint 2010 is answering this challenge with rich media capabilities. SharePoint 2010 comes with

- The infrastructure to support storing and serving media using the asset library template
- Asset library support for RSS/podcasting while still supporting standard document library features
- The ability to play videos within SharePoint using the media Web Part or using remote locations
- Support for customized skins using Expression Blend 3
- Access to playback controls using the JavaScript object model

This evolution of rich media support allows SharePoint 2010 to seamlessly incorporate videos, audio, and pictures into the collaboration experience without extensive training or support. For example, the SharePoint Media Player can be included on any page with a few clicks of the mouse. To insert a Microsoft Silverlight Web part, complete the following steps.

1. Click Page in the Site Action toolbar.
2. Click Edit in the Ribbon.
3. Under the Editing Tools panel, click Insert.
4. Click Video And Audio.
5. Click inside the new Web Part to configure player settings. See Figure 1-9 for Silverlight Player options.

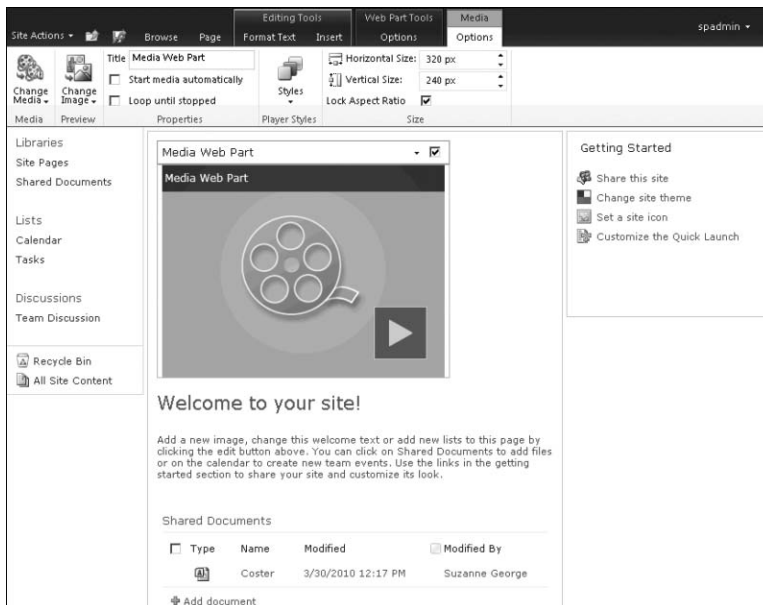


FIGURE 1-9 Silverlight Player configuration options

Working with Lists and Libraries

You add content to a SharePoint site using lists and libraries. Lists and their associated views provide a flexible mechanic for storing most kinds of information. Libraries, a special form of a list, usually store documents and their associated metadata. Lists and libraries provide users the ability to gain feedback instantly, get updates, and see historical information on data much more quickly and easily than using traditional methods of communication such as e-mail, network file shares, and so on.

SharePoint 2010, like previous versions, stores data in lists that are similar to Microsoft Excel spreadsheets. Much of the data consumed in SharePoint sites is located in lists, and each list item contains unique attributes. SharePoint 2010 offers standard templates to create lists as well as to provide the capability to create custom lists based on the structure and requirements of a user's application.

List Enhancements

SharePoint 2010 contains list enhancements to assist content administrators in gaining better control of their data. Configuration options and content control allow business users and content administrators to manage and ensure that fresh, relevant content to their users. Some of the list enhancements are

- New input form
- New list view
- Enhanced list lookup capabilities
- Referential integrity in the list—enforcing unique values
- Cascading deletes

List Input Form

One key to keeping data consistent in lists is the methodologies the users use to input data. If the user is required to upload data, remember links, and/or break information into separate pieces, each step paves the way to errors and input control issues. One change to lists that was frequently requested in the past has finally been added as an enhancement to SharePoint 2010: the ability to provide a single interface for adding list information, whether it is an image, document, link, or other textual information. Furthermore, the input screens, which were not always easy to use—especially for a beginner SharePoint user—have also been improved.

The new data entry forms for list items are much more user friendly. Gone are the difficult-to-edit HTML pages that defined user inputs; now an elegant pop-up form provides you with an easy method for user input. Figure 1-10 shows the new input form for SharePoint lists.

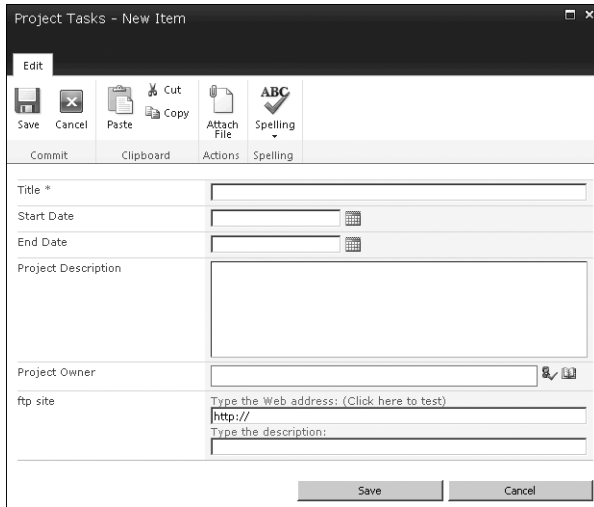


FIGURE 1-10 The new SharePoint 2010 list input form

The new Ribbon provides easy access to relevant information necessary for updating the list item. The Ribbon icons change depending on the context and the column type. For example, if you are working in a rich text field, you'll be presented with many familiar Microsoft Word functions, such as the Editing Tools tab on the Ribbon, shown in Figure 1-11.

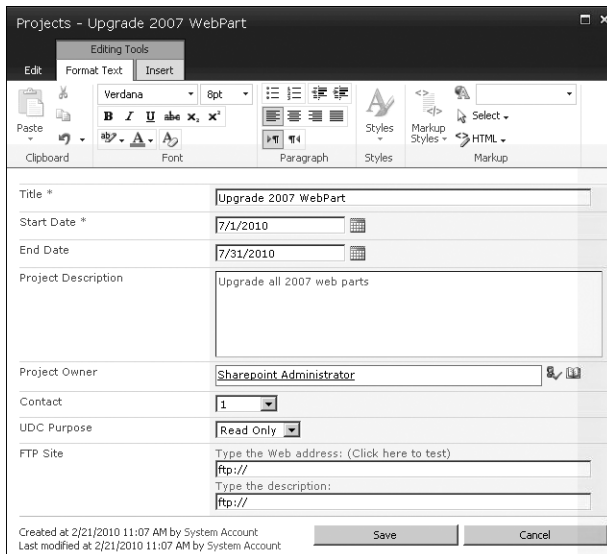


FIGURE 1-11 The Editing Tools tab on the Ribbon

Further, when you click the Insert tab under Editing Tools, the new forms that appear allow you to insert a table, picture, or link onto the page. You can also upload a picture directly into the rich text box to provide additional information as needed to describe the list item. The

process of adding graphic elements to a page has been streamlined—no longer do you have to leave the Add or Edit form to first upload an image, then find the URL, and finally paste it in the box. Figure 1-12 shows the interface presented when an image is uploaded. Note that SharePoint will automatically upload the image into the list you specified.

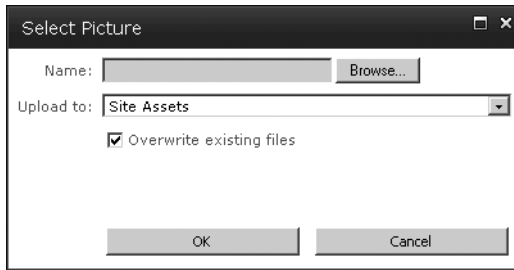


FIGURE 1-12 The Select Picture dialog box allows you to upload an image directly to the list you specify.

The seamless interface is continued, as shown in Figure 1-13, by prompting you for the title of the image and any other columns defined by the SharePoint library selected.

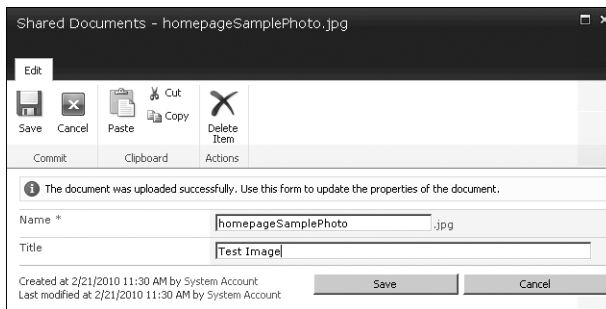


FIGURE 1-13 Set file properties for an uploaded image

After the file successfully uploads to the list, the List Edit page reloads and the Ribbon now contains a Picture Tools tab. This tab allows you to execute actions on the image, such as changing the picture, position, size, and URL information, as shown in Figure 1-14.

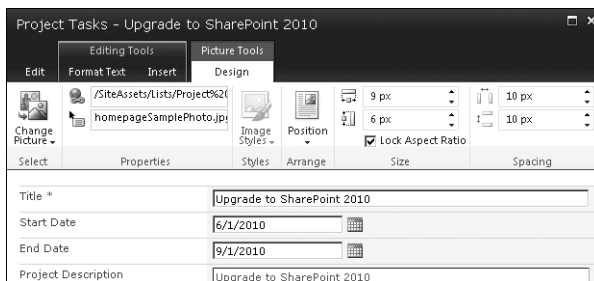


FIGURE 1-14 The Picture Tools tab lets you manipulate an image uploaded to a list.

Inserting a table is just as simple and also offers rich features you can use to customize the table. The methods are very similar to inserting a table in a Microsoft Word document, setting the rows and columns and selecting table styles and formatting elements. The Table Tools Layout tab on the Ribbon provides you with an array of table layout functions, shown in Figure 1-15.

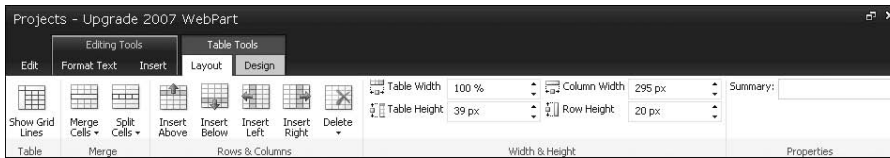


FIGURE 1-15 The Table Tools Layout tab on the Ribbon

The Table Tools Design tab features the Ribbon shown in Figure 1-16.

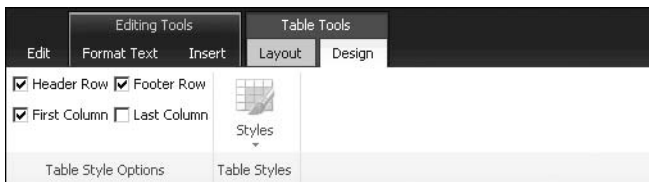


FIGURE 1-16 The Table Tools Design tab on the Ribbon

The method for formatting URLs has been improved as well. When you insert a URL using the Ribbon, you can type a description, open the link into a new tab, and display an icon. Figure 1-17 shows the Link Tools tab on the Ribbon, with options for formatting hyperlinks.

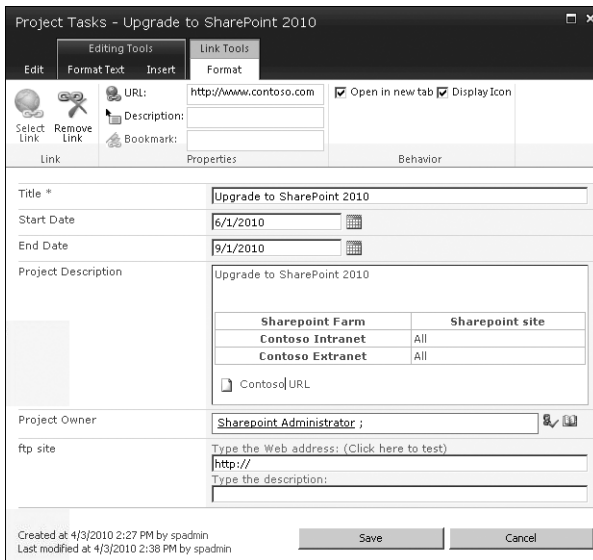


FIGURE 1-17 The Link Tools tab on the Ribbon provides options for formatting URLs as links on a page.

With SharePoint 2010, you have many options for entering information into lists, and with enhanced features such as the rich text box, you can clearly articulate the information requested by list content managers.

Another feature new to list form templates are the Add New and Edit forms. These forms are similar to each other, providing another consistent user interface. When you click Edit in the Ribbon, the Edit form opens and you can use it to perform data updates on the list item. When you edit an existing list item, you also can view more information about it to manage permissions, check version history (if available for the list), or delete the list item. The Edit form is shown in Figure 1-18.

The screenshot shows the 'Edit Item' form in SharePoint 2010. The title bar reads 'Project Tasks - Upgrade to SharePoint 2010'. The ribbon includes 'View', 'Edit Item', 'Version History', 'Alert Me', 'Manage Permissions', and 'Delete Item'. The form fields are as follows:

- Title:** Upgrade to SharePoint 2010
- Start Date:** 6/1/2010
- End Date:** 9/1/2010
- Project Description:** Upgrade to SharePoint 2010

Sharepoint Farm	Sharepoint site
Contoso Intranet	All
Contoso Extranet	All
- Project Owner:** Sharepoint Administrator
- ftp site:**

At the bottom, it shows 'Created at: 4/3/2010 2:27 PM by spadmin' and 'Last modified at: 4/3/2010 2:38 PM by spadmin'. A 'Close' button is located at the bottom right.

FIGURE 1-18 SharePoint 2010 Edit form

When you have updated the item, the list view will look like the screen shown in Figure 1-19. Notice that in the list view you see the full rich text of the list elements.

The screenshot shows the 'List Tools' ribbon with 'Items' and 'List' tabs. The breadcrumb path is 'Home > Project Tasks > All Items'. The search bar contains 'Search this site...'. The left sidebar shows navigation options like 'Libraries', 'Site Pages', 'Shared Documents', 'Lists', 'Calendar', 'Tasks', 'Project Status', 'Project Tasks', 'Discussions', 'Team Discussion', 'Recycle Bin', and 'All Site Content'. The main content area displays a list with the following data:

Title	Start Date	End Date	Project Description	Project Owner	ftp site
Upgrade to SharePoint 2010	6/1/2010	9/1/2010	Upgrade to SharePoint 2010	Sharepoint Administrator	

The 'Project Description' field in the list view contains the same rich text table as seen in the edit form:

Sharepoint Farm	Sharepoint site
Contoso Intranet	All
Contoso Extranet	All

Below the table, there is a 'Contoso URL' link and an 'Add new item' button.

FIGURE 1-19 Rich text list view

The seamless data entry process provided in SharePoint 2010 will reduce data errors and frustration levels for all users. Now you can easily enter data in a format that suits your needs,

and power users and content managers will have cleaner, more appropriate data with less expensive custom development efforts.

List Templates

When you create a new site, several lists are automatically available for use. The site developer should experiment with the different list types before determining which list template provides the best functionality and the features most needed. For example, the Team Site template will create Calendar, Tasks, and Status lists. Each of these lists will provide the user with different functionality.

As in previous versions of SharePoint, the custom list provides a good introduction to using and managing list data. When you create a custom list, it allows you to add views, columns, and items as needed, as shown in Figure 1-20.

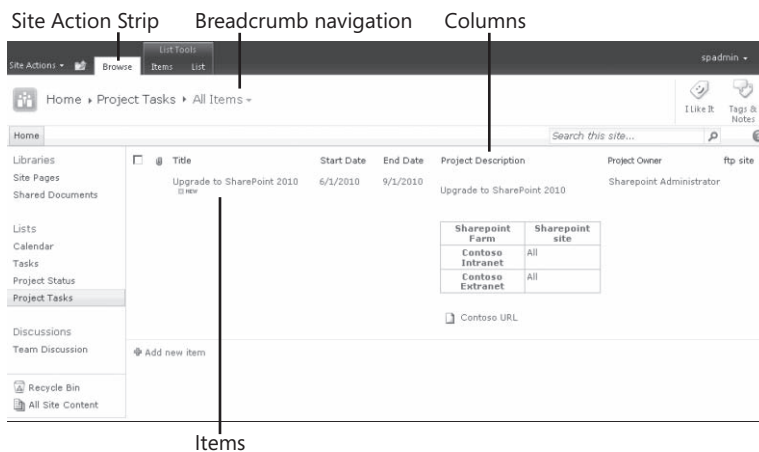


FIGURE 1-20 SharePoint site list view

- **Items** Rows of individual data components.
- **Columns** Define the types of data located in the row. Columns can be created to contain only certain information, such as e-mail addresses, URLs, and dates. Customized and lookup data types are also available. Columns such as Create/Modify Date and Create/Modified By are usually not shown by default; however, they are part of the list.
- **Site Action Strip** Allows a user or content administrator to access and perform different functions on lists. You can find functions such as changing views, adding columns, and exporting data here.
- **Breadcrumb navigation** As with previous version of SharePoint, breadcrumb navigation helps to define the user's location within the SharePoint Farm hierarchy. Having breadcrumb navigation available is important so users can quickly navigate within the various SharePoint sites. By default, breadcrumb navigation is replaced with the Ribbon as actions on content are available to the user.

The new Ribbon toolbars relocate common tasks and dynamically adjust based on your security role. The following images show what a content manager or administrator might see. Figure 1-21 shows the actions that can be completed on individual items in the list such as a document or list item.

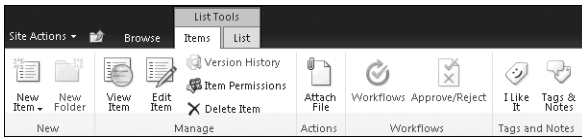


FIGURE 1-21 Available actions on the Items tab

Figure 1-22 shows the actions that can be done on the list itself. These actions allow the administrator or power user to modify and create views, export list items to Excel or a document in other Office applications, and change list settings.

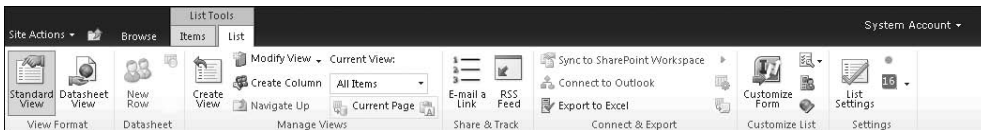


FIGURE 1-22 SharePoint 2010 Ribbon bars

Creating a List

The process of creating a list is very similar to creating a site. Several list templates are available; Figure 1-23 shows the available list templates using the new Silverlight menu controls. Each of the default templates shown in Figure 1-23 showcases different features that are available for use with list data.

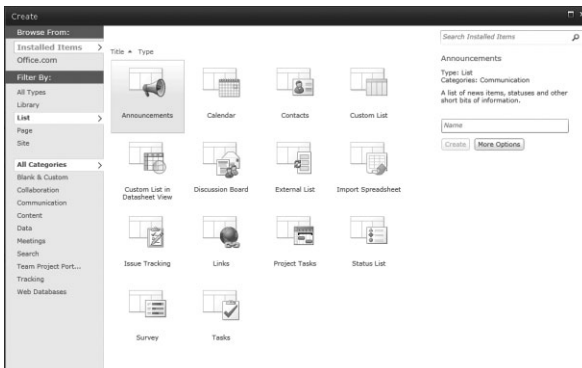


FIGURE 1-23 SharePoint 2010 Create List types

To create a custom list, a content administrator can go to the SharePoint team site and follow these steps.

1. Click Site Actions and then select More Options to open a new window with all the SharePoint site templates available (Figure 1-23).
2. Click the List link.
3. Choose the list type you want to create.
4. Type the name for the list and then click Create.

After you have created a list, you will see the new, empty list appear along with the task Ribbon bars described in the previous section. Now you can add data, columns, and views to the list.

Managing List Columns

The power of lists resides in creating columns that will help to provide users with the metadata they want and need. As when designing a SQL database or other business application, it is important to understand the types of metadata users will require and how the data relates to other corporate information. For example, if a corporation wanted to upload all project files into a SharePoint document library, metadata would be required to link the files together. Thus, a column would need to be created that would hold the project name or identifying number so a user would be able to find all documents relating to a specific project. Other columns might be created that would link information such as clients and business groups. All SharePoint columns have a specific type and validation for associated data. Columns can be added to

- Show text and other basic information
- Provide predefined choices for the user for enhanced grouping or filtering
- Link to other site data
- Calculate values based on data in other columns

Figure 1-24 shows the list of standard column types available in SharePoint 2010.

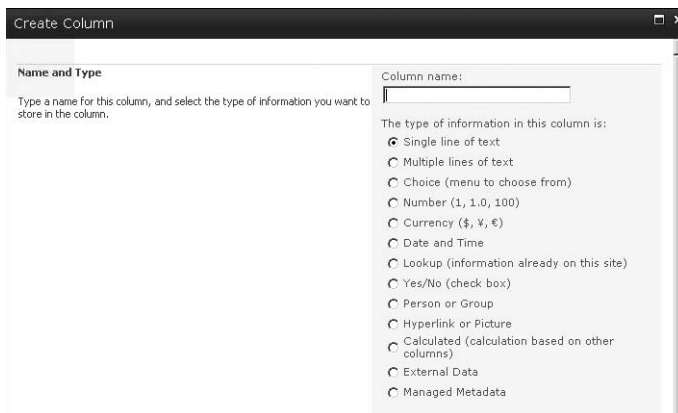


FIGURE 1-24 Standard column types available on the Create Column page

Creating, Editing, and Deleting Columns

You can insert columns to prompt a user to provide additional information for an item in the list. To add columns to a list, complete the following steps.

1. Open the list and click the List tab under List Tools.
2. Click the Create Column icon.
3. Type a short, descriptive name for the column.
4. Choose the column type for the information to be stored in the list.
5. Specify any additional settings as needed. The settings displayed are dependent on the column type and will dynamically change to reflect the column type you choose. You will find settings such as Required Information and Enforce Unique Values in this section.

NOTE If you choose the Enforce Unique Values option, you will be asked to index the column. It is important to click OK so SharePoint will index the column to improve system performance. Further, if you are modifying an existing field to make it unique, SharePoint will validate the data and return an error if the data already in the column is not unique.

6. Add column validation as needed. You can specify a formula that will validate the data in this column when new items are saved to this list. The formula must return a true or false validation and cannot refer to other fields in the list. Example: If a column is named Length Of Project, a valid formula would be *[Length of Project] > 0*.
7. Type a description for the validation formula and an example of what is needed for the validation to pass. Example column description: *Please enter the number of days the project will last: 40*.
8. Click OK to create the column.

When you have completed these steps, the list refreshes and returns the user context with the added column.

NOTE If you plan to use column names in queries or through a Web service, SharePoint Application Programming Interface (API), or other method, columns names should not contain spaces for better behavior within queries and API calls. If the column is created as LengthofProject and then modified to show Length Of Project, it will be possible to reference the column through the various APIs as LengthofProject, yet have it display to users with the spaces included.

To see a list of columns created for a list, click List Settings in the Ribbon. All columns in the list display. If the column name is selected, you can modify the column settings as needed. Be careful when changing a column type, because SharePoint will warn you that data might be lost. For example, changing a rich text box to a single line of text will result in SharePoint

dropping any data that is longer than the new type will support. The validation rules might need to be adjusted as well. SharePoint does not allow certain column types to be changed to other column types. If this is required, the columns must be deleted and re-created. Understanding the column data and its requirements avoids having to make column type changes to lists.

You can delete a column by clicking List Settings in the Ribbon, just as when you modify a column. When you select Delete, all the data in that column is deleted, and recovery is impossible using the SharePoint Recycle Bin.

IMPORTANT Use caution when deleting a column, because data in a deleted column cannot be retrieved later.

Types of Columns

There are three common column types in SharePoint 2010: information, lookup, and calculated columns. New column types such as external data and managed metadata will be discussed in Chapter 14, “Administering Enterprise Content Management.”

INFORMATION COLUMNS

The traditional information-gathering columns used in SharePoint, such as Single Line Of Text and Hyperlink Or Picture, allow the user to enter supporting information for the list item. For example, in a project list, supporting information could include a project description, the start and end dates of the project, and the project owner. Figure 1-25 shows an example of a new project list with these types of information columns.

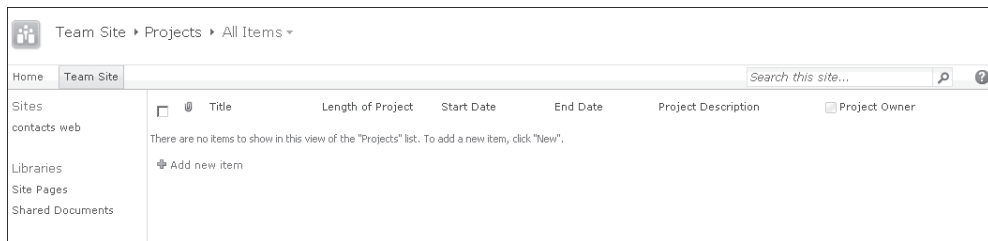


FIGURE 1-25 List view of a project list with information columns added

LOOKUP COLUMNS

Lookup columns, enhanced to show additional column information from the referenced list, allow a user to link list data together. To create a lookup column in list view, complete the following steps.

1. Click List under the List Tools tab.
2. Click Create Column to display the Column page.

3. Type the column name.
4. Choose Lookup (Information Already On This Site).
5. Set the additional column settings for the column as shown in Figure 1-26.

Notice the new options available in SharePoint 2010. Lookup columns are now able to show more than one column from the referenced list. To show additional column information from a lookup list, select the check box to Allow Multiple Values under the In This Column drop-down box. After you have selected the check box, columns will be created for each selected column metadata. This capability is important so the user can be presented with the necessary information with fewer clicks. In Figure 1-26, if a user needed the ability to view a contact's Mobile Number and Home Phone alongside the current list data, the Allow Multiple Values, Home Phone, and Mobile Number check boxes would all be selected.

Additional Column Settings
Specify detailed options for the type of information you selected.

Description:

Require that this column contains information:
 Yes No

Enforce unique values:
 Yes No

Get information from:

In this column:

 Allow multiple values

Add a column to show each of these additional fields:

- ID
- Last Name
- Modified
- Created
- Version
- Last Name (linked to item)
- First Name
- Full Name
- E-mail Address
- Company
- Job Title
- Business Phone
- Home Phone
- Mobile Number
- Fax Number
- City
- State/Province
- ZIP/Postal Code
- Country/Region

Add to default view

FIGURE 1-26 Creating a lookup column

6. Set the relationship by choosing whether to enforce relationship behavior when the list item is deleted. When an item in the target list is deleted, a *cascade* delete will delete all related items in this list. A *restricted* delete prevents deletion of a list item in the selected target list if one or more referenced items exists in the list. For example, a project list is created that details information about the projects in the engineering group. A secondary list, called Project Tasks, is also created, and it has a column lookup to the project list.
7. If the Enforce Relationship Behavior check box (shown in Figure 1-27) is selected and Restrict Delete is selected, the user cannot delete a project until all referenced Project Tasks have been deleted. If the Enforce Relationship Behavior option is selected and Cascade Delete is selected, as shown in Figure 1-27, all items in the Project Tasks will be deleted if the project list item is deleted.

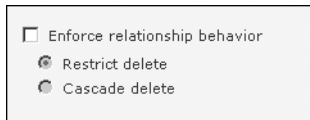


FIGURE 1-27 Enforce Relationship Behavior option

8. With Referential Integrity selected, you will be prompted to index the column. Click OK.
9. Click OK to create the column.

Figure 1-28 shows the list with an added lookup column.

Title	Length of Project	Start Date	End Date	Project Description	Project Owner	Contact	Contact:First Name
Upgrade to SharePoint 2010	60	6/1/2010	6/30/2010	Upgrade to SharePoint 2010	Sharepoint Administrator 1	1	SharepointAdmin

FIGURE 1-28 Default list view with added lookup column

The lookup column called Contact has a value of 1 in Figure 1-28, and the Contact:First Name column is also viewable in the list, thereby showing two columns from the referenced list for users. Since this lookup column was set to Cascade Delete when the column was created, this list entry will be deleted in the event that the user SharePointAdmin (the Contact:FirstName column entry) is deleted.

CALCULATED COLUMNS

Calculated columns bring content together from other columns in the list. In the section titled “Creating, Editing, and Deleting Columns” earlier in this chapter, you were asked to type the length of the project (see Figure 1-18). Since the project start and end dates are part of the list item metadata, the project length can be calculated by the system. A calculated column can be created by following these steps.

1. Open the list and click List under the List Tools tab.
2. Click Create Column to display the create Column page.
3. Type the column name.
4. Click Calculated Column.
5. Create a formula based on existing columns, as shown in Figure 1-29.
6. Click OK.

FIGURE 1-29 Adding a calculated column to a list

MORE INFO For additional information about how to use calculated values, navigate to <http://blogs.msdn.com/mcsnoiwb/archive/2008/05/04/sharepoint-formulas-in-calculated-columns.aspx>.

Indexing Columns

With SharePoint 2010, large lists are now a supported feature. SharePoint now enables sites to have virtually unlimited numbers of items in lists and libraries. However, the core of the SharePoint platform is Microsoft SQL Server, and performance is affected when large amounts of data are queried without the use of indexes. Therefore, it is important to index large lists to streamline overall performance.

When setting up indexes for large lists, consider the following points.

- Microsoft has increased the number of columns in a list index to 20 in SharePoint 2010.
- Each column index consumes resources in the database. Farm or content administrators should only add indexes to columns that will be actively used in views.
- Columns that have been set to allow multiple values cannot be indexed and will not be listed.

To add an index to a site, complete the following steps.

1. Click List under the List Tools tab and then select List Settings.
2. Click Indexed Columns.
3. When the current indexes on the list are shown, click Create A New Index.
4. Click the column that you want to use for the primary column in the index.
5. Select the secondary index column. If this field is left blank, the resulting index will be a single column index. If a second column is specified, the resulting index becomes a compound index in which only certain fields can participate.
6. Click OK.

Sorting and Filtering

Sorting and filtering list data, especially when there are a large number of items in a list, is beneficial not only for SharePoint performance as a whole, but also in assisting you in finding information quickly and easily in a list.

Lists in SharePoint 2010, like previous versions, allow users to sort and filter data using techniques generally found in other Microsoft Office applications. Each heading in the list view allows you to sort or filter the data based on your needs or requirements. For example, if you want to sort your tasks by date, a simple click on the Date column header to reformat the list data sorted by date. By clicking the Date column header again, the data in the column will be sorted in reverse date order, and you will see the list refreshed to show it in that order in the new list view. As shown in Figure 1-30, when the arrow in the header bar is pointing up, the list is sorted in ascending order; when the arrow in the header bar is pointing down, the list is sorted in descending order.

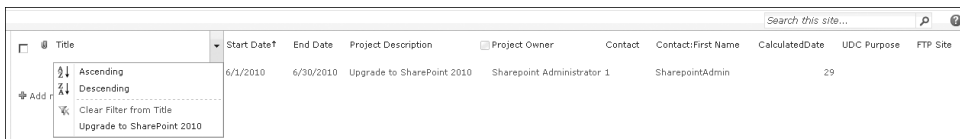


FIGURE 1-30 List sorting and filtering

Filtering, on the other hand, differs from sorting in that it allows you to view only the data in a list that meets specific criteria. Filtering lets you to sift through large amounts of data to find only the data you need, based on the information contained in a column you select. If you want to view all tasks due today, for example, click the arrow next to the Date Due column heading, and a drop-down list displays the available dates to use as a filter for the

list. When you select today's date to use as a filter, only the tasks in the list due on that date will appear in the list view. You will see a funnel-shaped icon displayed in the column header when filtering is enabled.

IMPORTANT Not all columns support filtering. Notable columns are Multiple Lines Of Text and hyperlink.

List Views

Although user sorting and filtering are useful, an even more powerful feature is list views, in which you can use predefined sorting and filtering to determine what you see in the list—you do not need to resort or add a filter each time you view the list. List views allow users, content managers, and administrators to prefilter and/or sort lists to provide the user with only data relevant to the user. List views describe the data, columns, sorting of data, and the style used to present the data. Views, unlike traditional reports, are dynamic and refresh every time the page loads.

List views can be either public or private, meaning that the content manager or administrator can set up predefined views for all users to see, or a user can create a personal view that describes the list data in a way that suits his or her specific needs at the time. All of the sorting and filtering preferences can be saved for use later.

Farm and content administrators should consider the following points when working with list views.

- Only one indexed column can be used in a view filter. Even though filtering on different columns is possible, the first column in the view should be indexed to reduce the overall number of items returned.
- A view can encompass more than one search criteria; however, any view that uses an OR filter does not have any index benefits described here.
- Defining an item limit in the view does not provide the same results as a filtered view.
- You should always use a filter that will return no more items than needed. If a list has 10,000 items, for example, but 9500 of them are not needed daily, add a filter to the default view.

To create a list view, complete the following steps.

1. Click List under the List Tools tab and then select List Settings.
2. Click Create View.
3. Choose either a view format or the option to start from an existing view.
4. Type a view name.
5. Choose either a public or personal view.
6. Choose the columns and set the order in which the columns will be displayed.

7. Choose the sorting method needed—if none is chosen, the list will sort by ID.
8. Choose the columns to filter.
9. Choose whether to allow inline editing. If inline editing is enabled, each row will have an Edit button that enables edit features without requiring the user to leave the view. Inline editing is only available when the style is set to Default.
10. Choose whether to allow tabular view. These check boxes allow you to select multiple items in a view and perform bulk actions such as Delete.
11. Choose the columns to group by.
12. Choose totals, if needed.
13. Change the style for the view if needed.
14. Display all items with or without folders.
15. Choose an item limit to limit the amount of data that is returned to users of this view. You can either make this an absolute limit or allow users to view all the items in the list in batches of a specified size.
16. When a view has been enabled for mobile access, it is also possible to make the view the default view for users of mobile devices. This allows content views to be tailored for both the PC (which may have additional columns and/or graphics) and for mobile users, providing the latter with a simpler, lightweight view. When a mobile user views the list, she can click the key field (specified in the view in mobile settings) to view the entire content record.
17. Click OK.

When the view is created, the page is refreshed and the new view appears. To change the view back to the default, click the List tab in List Tools and click the arrow under Current View, as shown in Figure 1-31; then select Default from the drop-down list.

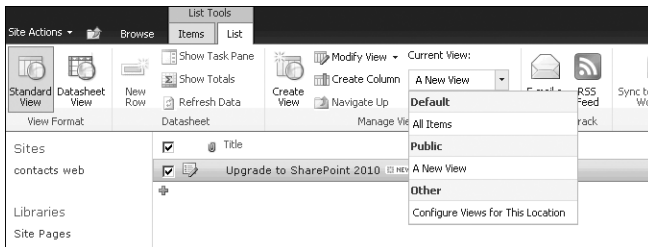


FIGURE 1-31 Changing the list view

To see the inline editing in action, select the check box to the left of the list item you want to edit. SharePoint highlights the row and the Ribbon changes to the Items action tab under List Tools, as shown in Figure 1-32. A new Edit button appears to the left of the first column. If you click the Edit button, you can list items within the view.

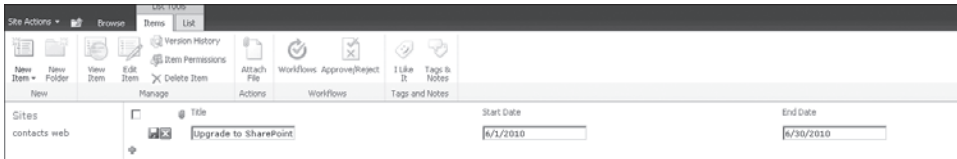


FIGURE 1-32 Edit list items within the view

To save the changes you make, click the blue Save icon. To cancel the edits you made, click the red X icon. To add new items using this view, click the green plus sign (+) icon found below the last item in the list.

Advanced List Concepts

Additional features in SharePoint 2010 lists allow for additional customization and administration of list data. By providing support for large lists, site columns, and advanced input forms, site administrators are able to control not only the data into the list, but server resources, global farm access to list data, and the methods users use to enter data into lists.

Large List Support

SharePoint has traditionally been able to support large lists; however, the challenge has always been to make the data in large lists available for retrieval and display in a timely and efficient manner. Although a user can create millions of items in a SharePoint list, the query operation slows as the size of the list grows. A new addition to SharePoint 2010 is large list support throttling. Throttling allows farm administrators to set manageable limits on the number of items queried from a SharePoint 2010 list. Using throttling, SharePoint 2010 encourages developers and farm administrators to control large list queries to improve the performance of a SharePoint farm.

Throttling thresholds are set on site collections. By default, standard users are throttled at 5000 items returned from a list, with a warning threshold at 3000 items. (See Table 1-6 for a list of items that would be throttled.) These values can be changed using Windows PowerShell commands. The throttling mechanic works as follows: If a user is pulling back a list with 4999 items, the query returns all of the data without error; however, if the user or query tries to retrieve 5001 items, an expensive query exception will be returned. The following parameters are set by default.

- Warning for user throttling at 3000 items
- Standard user throttling at 5000 items
- Super user throttling at 20,000 items

NOTE Developers can request a throttle override from the Object Model by using *SPQuery.RequestThrottleOverride* and *SPSiteDataQuery.RequestThrottleOverride*, which will effectively cause SharePoint to ignore the throttling behavior.

TABLE 1-6 Items Throttled for a Standard User with Large List Support Throttling

NUMBER OF ITEMS RETURNED	QUERY RUN IN NORMAL LIST AS STANDARD USER
<5000 list items	Code and/or view query will run and return the required result.
>3000 list items	No exception is thrown and the code will run properly.
>5000 list items	An expensive query exception is thrown.
>5000 list items during time window OK for large queries	Query threshold is not applied, query runs even if the threshold has been exceeded.
Object Model Override = true and RequestThrottleOverride is overridden	Exception is thrown if list is throttled, and this property cannot be overridden by a normal user.

Using Central Administration, a farm administrator can configure a time frame in which these expensive queries can and cannot be run. Therefore, if a user wants to run a query with 5001 items at 5:30 P.M. and the time-frame limitation is between 8:00 A.M. and 5:00 P.M., the query would succeed. To see if a list is being throttled or within warning limits, go to List Settings and look at the bar that gives the number of items in the view/list.

To find out the current throttling limits, follow these steps.

1. Open a browser and go to the SharePoint Central Administration website.
2. Under Application Management, select Manage Web Application.
3. Select the Web application that you want to modify or view.
4. Click General Settings in the Ribbon and select Resource Throttling. Figure 1-33 shows the Central Administration Resource Throttling command on the General Settings menu.

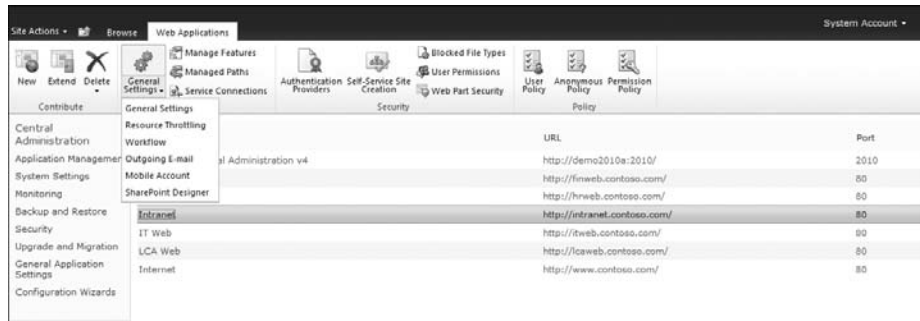


FIGURE 1-33 Resource Throttling

Figure 1-34, which displays list resource throttling parameters, is the first of several pages of information you will see with resource throttling settings. A farm administrator should be familiar with these settings before any changes are made to a SharePoint site. Also, it's im-

portant to test all new settings in a development environment before applying configuration changes in production.

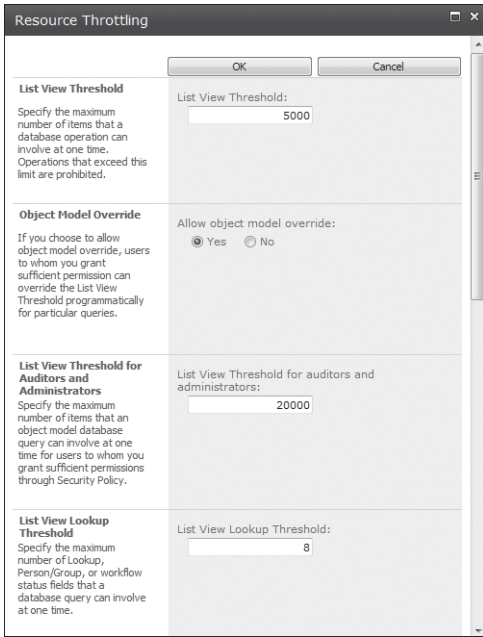


FIGURE 1-34 List Resource Throttling parameters

An Upgrade Warning: Setting Throttle Limits

When upgrading from SharePoint Server 2007 to SharePoint 2010, any list that contains a default view of more than the throttled limit (default: 5000) will not be immediately available until a new default view is created.

A farm administrator should become familiar with the Windows PowerShell commands, which will display and set the throttling limits. These commands are listed in Table 1-7, with examples.

TABLE 1-7 Windows PowerShell Commands for Displaying and Setting Throttling Limits

COMMAND DESCRIPTION	EXAMPLE COMMAND
Get max threshold values for Web application	<pre>\$sitecol = Get-SPSite http://mysharepointserver:5000 \$sitecol.WebApplication.MaxItemsPerThrottledOperationWarningLevel \$sitecol.WebApplication.MaxItemsPerThrottledOperation \$sitecol.WebApplication.MaxItemsPerThrottledOperationOverride \$sitecol.WebApplication.AllowOMCodeOverrideThrottleSettings</pre>

COMMAND DESCRIPTION	EXAMPLE COMMAND
Get max threshold values for Web application— example usage	<pre>\$sitecol = Get-SPSite http://mysharepointserver:5000 \$sitecol.WebApplication.MaxItemsPerThrottledOperationWarningLevel = 2000 (set throttle warning to 2000 items – default 3000) \$sitecol.WebApplication.MaxItemsPerThrottledOperation = 4000 (set throttle to 4000 – default 5000) \$sitecol.WebApplication.MaxItemsPerThrottledOperationOverride = 30000 (set super user throttle to 30000 – default 20000) \$sitecol.WebApplication.AllowOMCodeOverrideThrottleSettings = False (set code override for throttling to False – default True)</pre> <p>Note: Values in this section are for example purposes and are not recommendations</p>
Find large list throttle setting	<pre>\$siteCollection.RootWeb.Lists["LargeListName"]</pre> (look for IsThrottled in results)

In summary, SharePoint 2010 large list support will support lists with

- Up to 50 million items—during read operations
- Configuration options for administrators per Web applications
- Privileged operational override support for users with site collection or list administration privileges
- Default query restrictions
- Throttling when a list reaches a predefined of items

Site Columns

In traditional database application development, calls are often necessary that refer to a set of table data multiple times during the application execution. By having a single authoritative source for data, applications can ensure consistency across all application modules. This is often true in SharePoint as well—to keep consistency across the SharePoint farm, a cross-site lookup column is required in some cases. Cross-site lookups cannot be done using a standard lookup column, however, so a site column should be used instead.

Site columns are defined at the site collection level and are accessible to all sites under the site collection. A few common examples of standard SharePoint site columns are FTP Site and Gender. The FTP Site column is a URL field in which a URL and description can be entered; a Gender column provides the user a choice of Male or Female. In both cases, there is no ambiguity about what the user will need to type across all SharePoint sites. The primary advantage of using a site column is its ability to provide a consistent user response for the column.

To link to an existing site column, complete the following steps.

1. Click List under the List Tools tab and then select List Settings.
2. Click Add From Existing Site Columns.
3. Select the site group All Groups.
4. Select the FTP Site column from Available Site Columns.
5. Click Add and then click OK.

Creating a Site Column

To create a new site column in SharePoint 2010 from the root of the site collection, complete the following steps.

1. Click Site Actions in the Ribbon and select Site Settings.
2. Click Site Columns under Galleries.
3. Click Create. You will see a new page that you use to create the column, setting the title, settings, and validation as necessary. When you have finished selecting the settings for the new site column, click OK.

NOTE If, as the site administrator, you plan on creating many new columns, you should consider creating a new Site Column Group.

Modifying a Site Column

To modify an existing site column in SharePoint 2010 from the root of the site collection, complete the following steps.

1. Click Site Actions in the Ribbon.
2. Click Site Columns under Galleries.
3. Click the site column you want to modify.

IMPORTANT As with lists, only certain changes to the list type are supported, and content might be lost when converting from one type to another, such as going from a multiple lines of text type to a single line of text.

4. Change or edit the group if needed.
5. Change settings as needed.
6. Set the Update All Columns Based Upon This Site Column option to Yes to update all of the site columns. This operation can take a while to run, and any customizations to child list settings might be lost.
7. Click OK.

Editing List Input Forms

As with most applications, managers often request at least one feature that generally requires costly customization. A software platform can provide a thousand features, but one department's managers will focus on a customization they think they must have but which is either not available or is cost prohibitive to provide.





Prior to SharePoint 2010, the process of customizing list input forms was a grueling, cumbersome task and required SharePoint Designer or a custom Web Part. SharePoint 2010 makes editing the input forms easier by utilizing the InfoPath platform. The Ribbon for the list includes a section called Customize List, shown in Figure 1-35.



FIGURE 1-35 List Ribbon

There are four icons in this section, which assist the content or developer in modifying the input forms. Table 1-8 lists these icons with descriptions of each.

TABLE 1-8 Icons for Modifying Input Forms

ICON	DESCRIPTION
 Customize Form	The Customize Form icon allows a content administrator to use Microsoft InfoPath 2010 to create custom forms to add or edit items in the list.
	The Modify Form Web Parts icon allows a user to edit the default form Web Part and Web page using a Web browser interface.
	The Modify Form In SharePoint Designer icon quickly opens the form in SharePoint Designer 2010. You use SharePoint Designer to edit list settings, add or remove columns, and create new views, forms, workflows, and custom actions.
	The Create Quick Step icon allows you to create a custom Ribbon button to perform a custom action on a list. Creating a Quick Step will add a button to the Quick Steps group on the Items tab.

Modify Form Web Parts

You can make traditional Web Part changes to a new list form or edit the form within the Web browser. A content administrator can click the Modify Form Web Parts icon and select the appropriate form to edit. In the following example, the steps show you how to edit a new form and rearrange the Web Parts on the screen.

1. Click the Modify Form Web Parts icon and select Default New Form.
2. The Web Part property page opens, as shown in Figure 1-36, and you can use its options to make changes to the appearance, layout, and other advanced features. Under Advanced Features, a new option exists called Show Toolbar With Ribbon.

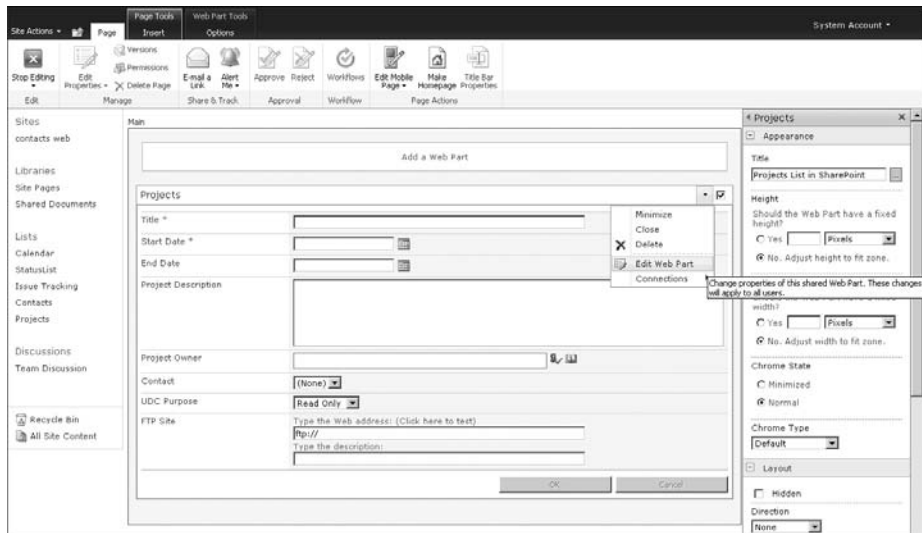


FIGURE 1-36 The Edit Web Parts property page

3. When you are finished making changes to the Web Part, click OK.
4. In the Ribbon, click Stop Editing.

You follow the same process to modify the Edit Page Web Part and the Default Display Web Part. For more information about managing Web Parts, see Chapter 19, “Web Parts and Their Functionality in SharePoint 2010.”

Modify Form Using InfoPath 2010

Microsoft InfoPath 2010 can now be used to modify the list forms. Within a few minutes a new, customized form is created, which allows form content to be grouped logically depending on form requirements. In SharePoint Server 2007, InfoPath forms were cumbersome to create and maintain; however, with SharePoint 2010, modifying the look and feel of a form to comply with user requirements is easy. For example, generally when a start and end date is required on a form, it is easier for the user to fill out the form when the two date columns are side by side. The following process explains the steps you would use to move the End Date next to the Start Date in a list form.

1. Go to the SharePoint list and click the Customize Form button in the Ribbon.
2. Microsoft InfoPath 2010 will open to allow edits to the new form. Figure 1-37 shows a default list edit form. In the right pane you see the fields available to place on the list. Making edits in the form is as easy as creating a new row or column and dragging fields to the form with the mouse.

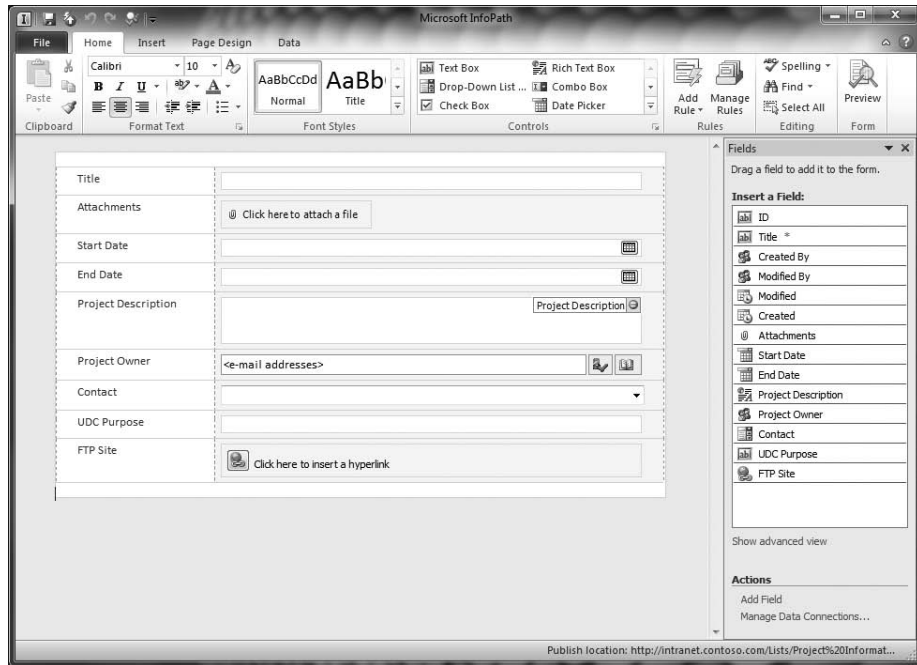


FIGURE 1-37 The default list edit form in InfoPath 2010

3. Make any additional changes to the page design by clicking the Page Design tab, as shown in Figure 1-38. You can modify the colors, fonts, and styles used in the page design as needed or required, similar to how you modify a spreadsheet in Excel.
4. When you have completed the changes you want to make, you should save the template and verify it for validity. To verify, click the File tab at the top of the page to open the Form Information panel. Click the Design Checker icon as shown in Figure 1-39 to verify the form contents and layout. After you run the Design Checker, control is returned to the form view (shown in Figure 1-37) and any errors are listed in the right pane. Fix the errors identified by the Design Checker, if there are any, and then return to the File tab to save and publish the form. If you are publishing more than one form type to a list or library, use Save As so the other forms are not overwritten.

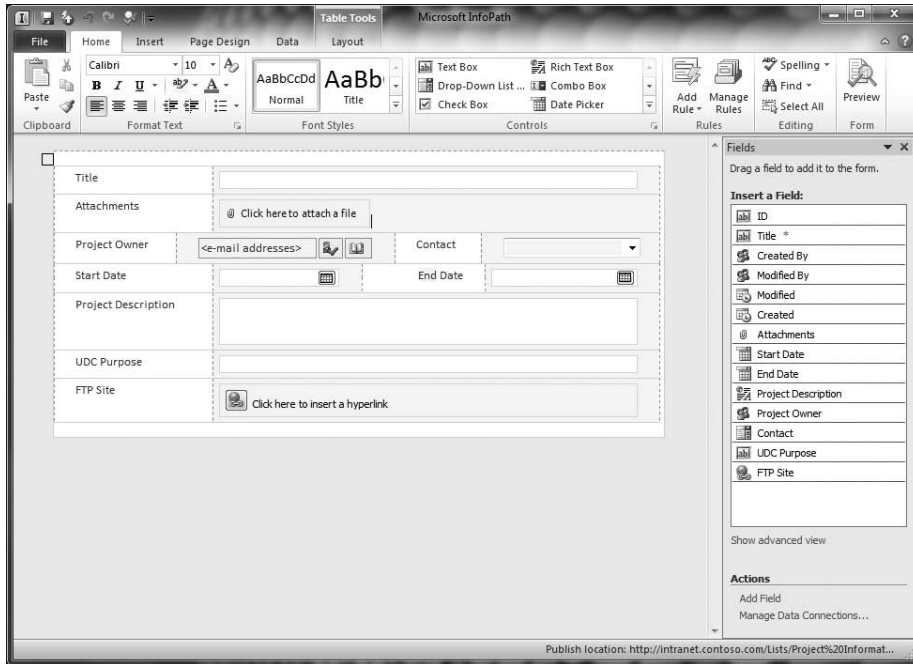


FIGURE 1-38 Changing the page design using the default list edit form in InfoPath 2010

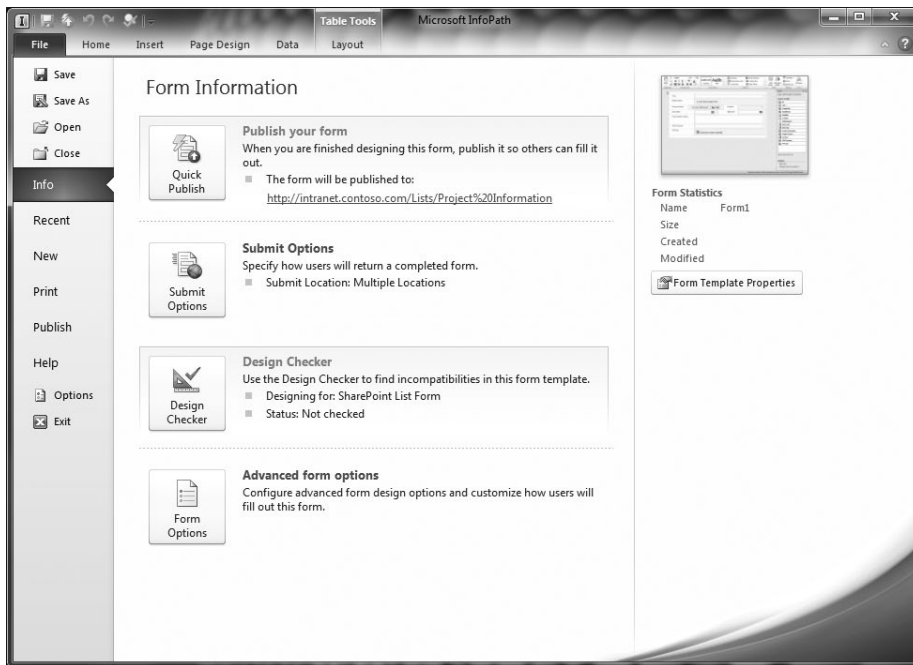


FIGURE 1-39 Design Form Page in InfoPath 2010

5. Publishing the form to SharePoint is as easy as clicking Publish Your Form. The SharePoint site is updated and all references to the Edit Form will now refer to the newly created list template. An example modification to the End Date field is shown in Figure 1-40. Notice that the Start Date and End Date fields are now located in this Project Information - New Item form.

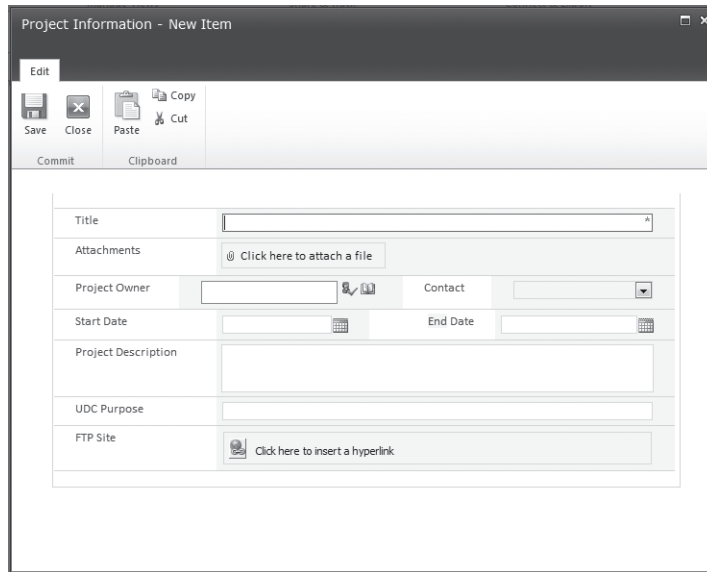
The image shows a screenshot of a SharePoint form titled "Project Information - New Item". The form has a ribbon at the top with "Edit" selected, and buttons for Save, Close, Paste, Cut, Commit, and Clipboard. The form fields are: Title (text box), Attachments (button: "Click here to attach a file"), Project Owner (text box with user icons), Contact (dropdown menu), Start Date (calendar icon), End Date (calendar icon), Project Description (text area), UDC Purpose (text box), and FTP Site (button: "Click here to insert a hyperlink"). The Start Date and End Date fields are highlighted with a light blue background.

FIGURE 1-40 Newly published form results

State Service Error Message

If you see the error message when you try to publish a modified form, the SharePoint 2010 State Service is not created:

"The form cannot be rendered. This may be due to a misconfiguration of the Microsoft SharePoint Server State Service. For more information, contact your server administrator."

One possible cause for this error could be that the SharePoint 2010 farm was created manually without running the farm Configuration Wizard.

To configure the SharePoint 2010 State Service using Windows PowerShell commands, complete the following steps.

1. Open a browser and go to the SharePoint Central Administration website.
2. On the taskbar, click Start, select Administrative Tools, and then select Windows PowerShell Modules. This will start Windows PowerShell and load all associated SharePoint modules.

continued on the next page

- 3. In the Windows PowerShell window, create a service application by typing the following command.**

```
$serviceApp = New-SPStateServiceApplication -Name "State Service"
```

- 4. Create a State Service database and associate it with a service application by typing the following command.**

```
New-SPStateServiceDatabase -Name "StateServiceDatabase"  
-ServiceApplication $serviceApp
```

- 5. Create a State Service Application Proxy and associate it with the service application by typing the following command.**

```
New-SPStateServiceApplicationProxy -Name "State Service"  
-ServiceApplication $serviceApp -DefaultProxyGroup
```

This will create the State Service service application, and InfoPath published forms should render properly. If you still see error messages when running custom forms, verify that the Web application is associated with the State Service service application.

Modify Form Using SharePoint Designer 2010

SharePoint Designer 2010 enables power users and developers to customize solutions—with little or no coding—that work for a variety of common scenarios ranging from collaborative sites and Web publishing to line-of-business integration, business intelligence, and human workflows. In this latest release, SharePoint Designer 2010 has been greatly enhanced and it is important to understand the changes in the new version. Features such as Contributor Settings, Database Interface Wizards, and Site Publish, Backup, and Restore no longer are available in SharePoint Designer 2010. For a full list of features that have been removed, refer to <http://technet.microsoft.com/en-us/library/cc179083.aspx>. Additionally, SharePoint Designer 2010 offers new features such as the ability to create Web Part Pages, master pages, lists, and workflows; setting permissions for individual users; and saving and deleting site templates, to name just a few.

In addition to the changes mentioned, you can extend the SharePoint 2010 Ribbon using standard development APIs to fulfill application and corporate requirements using SharePoint Designer 2010 or Microsoft Visual Studio 2010. You can customize both Ribbons and drop-down lists to display site lists, available Web Parts, workflows, or other information as needed.

As in other Microsoft Office 2010 products, Quick Steps can be created to execute a mini-workflow process. To create a Quick Step in the Ribbon that will update the full name of a contact if one hasn't been entered, complete the following steps.

1. Create a new contact list or open an existing contact list and click List under List Tools in the Ribbon.
2. In the Customize List section of the Ribbon, you will see icons that reflect the different actions available to customize the form. Refer to Table 1-7 for a description of the icons and their functions. To create a Quick Step, click the New Quick Step icon.
3. SharePoint Designer 2010 will automatically open and prompt you for the SharePoint site credentials. Enter an appropriate login and password.
4. SharePoint Designer will then ask if you want to start a new SharePoint workflow or an existing workflow on the Add A Button page, as shown in Figure 1-41.

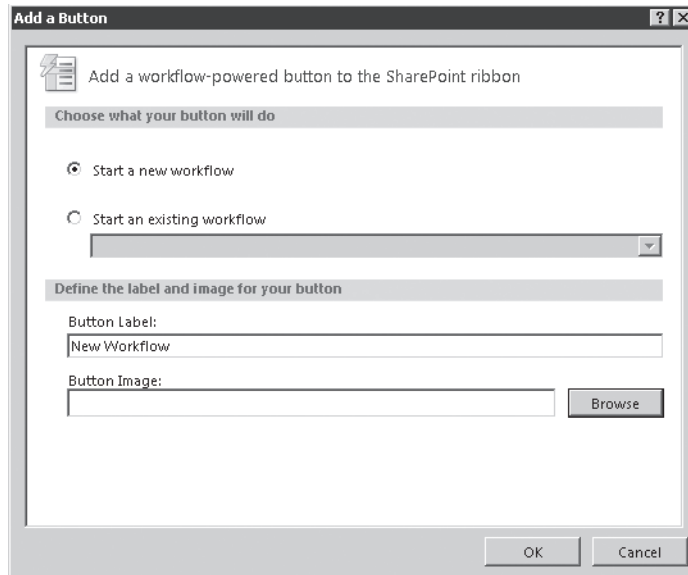


FIGURE 1-41 Using SharePoint Designer 2010 to add a new SharePoint workflow

5. Select the option to Start A New Workflow.
6. In the Button Label text box, type a label for the button you want to create.
7. Choose an image from the site. For example, you could use the Recycle Bin image at `/_layouts/images/fgimg.png`. You can click the Browse button to look for other images or type a path to the image you want to use directly in the Button Image text box.

NOTE Be sure to enter the image URL as a reference to the SharePoint site, such as `/images/imagefilename.png`; do not hard code the URL prefix by using a URL such as `http://www.sharepointsite.com/images/imagefilename.png`. This will ensure that all users will have access to the image, no matter the site collection or URL by which the page is called.

8. Click OK.
9. The new Button Design page displays. The controls on this page will be similar to Microsoft Visual Studio Workflow. Using the workflow example as a guide, shown in Figure 1-42, create the workflow steps using the Condition, Action, and Step tabs in the Ribbon. This workflow will update the full name of a contact if one hasn't been entered on a list item.

MORE INFO For more information on creating workflows in SharePoint Designer 2010, refer to <http://www.bing.com/videos/watch/video/getting-started-sharepoint-2010-creating-workflows-with-sharepoint-designer/10ts7flgd>.

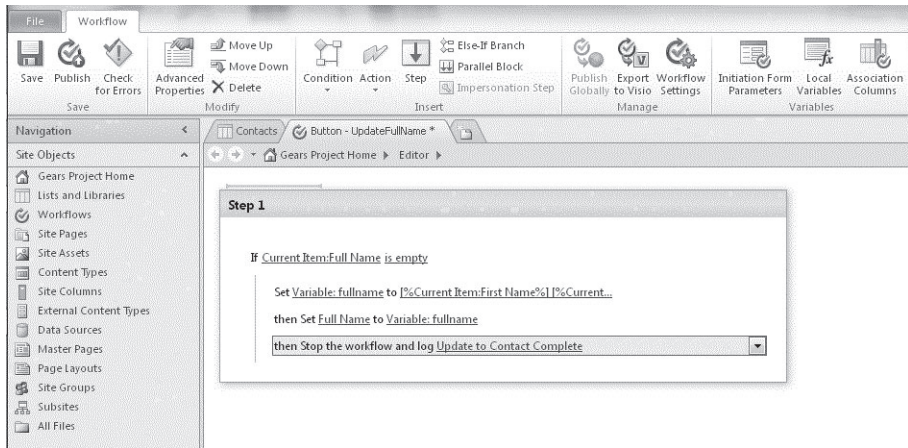


FIGURE 1-42 SharePoint Designer Add actions to Workflow

10. Save and publish the workflow as a new Quick Step. You will see the new Quick Step icon in your list when you click the List Tools – Items Tab. The new Quick Step icon will be located in the Quick Steps section of the Ribbon, as shown in Figure 1-43.

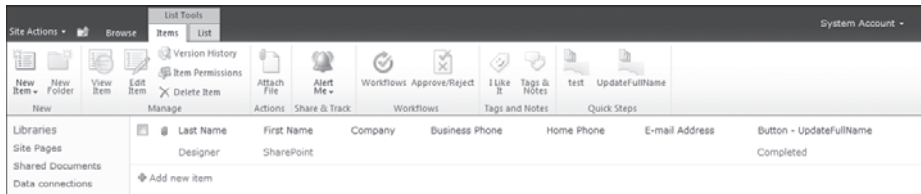


FIGURE 1-43 SharePoint Designer List with new Quick Step

11. To execute the new Quick Step, select the check boxes to the left of the list items you want to update and then click the new Quick Step icon. After the Quick Step has completed, a new field in the list will appear. In Figure 1-43, the new column name is called Button – UpdateFullName.

Microsoft Office Integration

SharePoint 2010 is more closely tied to the Microsoft Office application suite than previous versions. Along with the traditional Office integration features such as editing, check-out, and version control on Office documents, integration has been extended to include retrieving data from SharePoint lists and libraries. Integration to enterprise line-of-business (LOB) data is also available through the SharePoint Business Connectivity Services (BSC). More information on the BSC can be found in Chapter 18, “Aggregating External Data Sources.”

SharePoint Workspace 2010

One of the more interesting Office integration components is the Microsoft SharePoint Workspace 2010, which allows you to take SharePoint sites offline but maintain edit capability. SharePoint Workspace 2010 is a more evolved form of the product called Microsoft Groove, which allowed users to synchronize data and files between different systems. Many of the features in the Groove application still remain; however, those features are now adapted to include the SharePoint 2010 framework. SharePoint Workspace 2010 is also included with Microsoft Office Ultimate 2010 and is a must have for any users who are mobile and out of the office on a regular basis.

When SharePoint Workspace 2010 has been installed on the client system, it can begin synchronizing to SharePoint 2010 sites. Depending on the size of the SharePoint sites that need to be synchronized, you should probably recommend to users that they connect their laptop or desktop computer to the corporate local area network (LAN) during the synchronizing process to ensure that all content is initially downloaded successfully. The user’s computer must have adequate storage space, of course, to synchronize the SharePoint sites successfully.

The first time you execute the SharePoint Workspace 2010 application, you will need to create an account profile. When you complete the profile, the Launchbar is loaded (refer to Figure 1-45 for an idea of what the Launchbar looks like), and you will then need to attach to the SharePoint 2010 Server by completing the following steps.

1. Click New on the Home tab.
2. Type the SharePoint 2010 server name, such as **http://myservername**.
3. Click OK when asked if you want to synchronize the site.
4. Synchronizing will begin automatically, and when it has completed, you will see a notification page, similar to the one shown in Figure 1-44, that indicates the status of the synchronization process. You might find that there are some sites, lists, or libraries that are not supported yet.

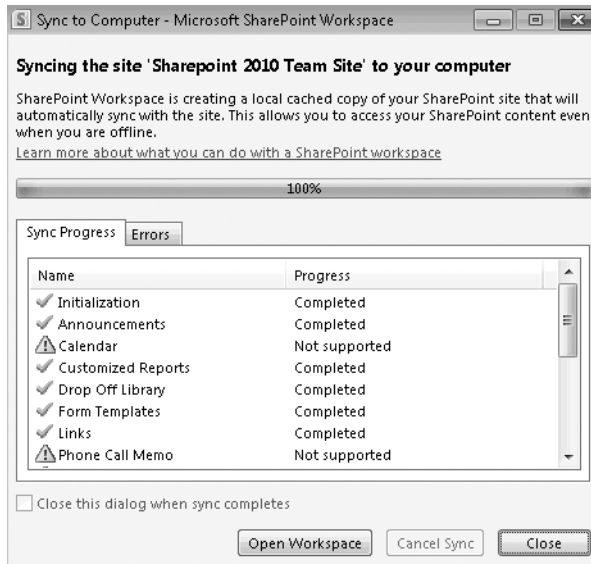


FIGURE 1-44 SharePoint Workspace synchronization notification page

5. At this point, SharePoint Workspace 2010 is now working online and will synchronize a single site at a time. If more than one site needs to be synchronized, the user must return to the initial SharePoint Workspace 2010 launch menu and create another link to a SharePoint site. The SharePoint Workspace Launchbar is shown in Figure 1-45. All sites that have been synchronized will be shown here.

When the site has finished synchronizing, there are a wide variety of actions you can perform on a site from within SharePoint Workspace 2010. For example, you can search items synchronized to the desktop from the Launchbar just as if you were accessing the SharePoint site in a browser. You can also set up alerts to notify a user of changes in the SharePoint site content.

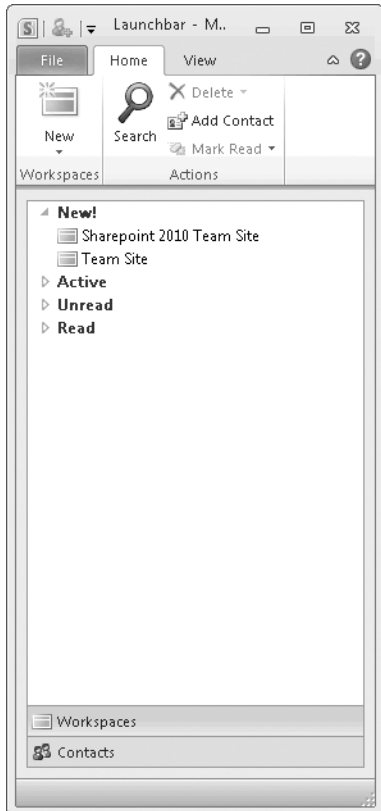


FIGURE 1-45 The SharePoint Workspace Launchbar

To add a site contact, click the Add Contact button in the Ribbon; to delete a contact, highlight the contact in the main panel and press the Delete key on the keyboard.

To open or work on documents in a synchronized site, click the SharePoint site name from the Launchbar. The SharePoint site will open and provide access to the available lists, libraries, and other site content. Figure 1-54 is an example of a SharePoint team site synchronized to SharePoint Workspace 2010.

In Figure 1-46, content is available for the user to access either online or offline. This functionality allows a user to synchronize a SharePoint site to a laptop computer and access documents even when disconnected from the corporate network. Certain list types and pages are available only if the workspace is connected to the SharePoint site, however. These lists and pages are clearly identified in the workspace client. In Figure 1-46, the lists in the Available On Server column are only available when the user is connected to the corporate LAN.

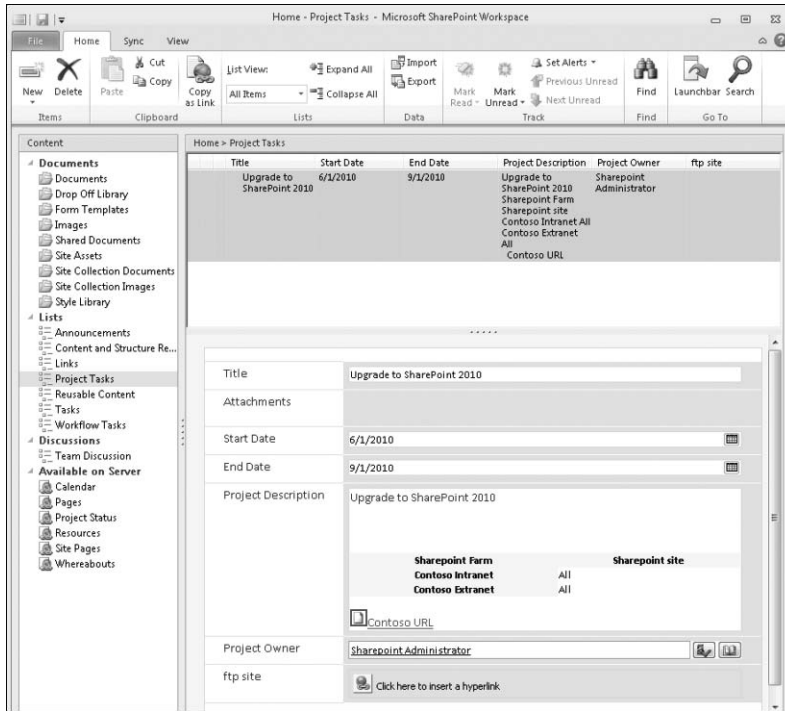


FIGURE 1-46 A SharePoint Workspace 2010 team site

The Project Tasks list in the example in Figure 1-46, however, is available offline and utilizes a customized InfoPath form for data entry. These forms are available offline, so users can enter data and synchronize it back to the server without error. Just as the SharePoint Ribbon changes based on field types, the Workspace client also changes. Some of the Ribbon items, such as workflow and custom Ribbon icons, are not available in the workspace; to use them, a user must connect to SharePoint Workspace 2010 using a Web browser.

In Figure 1-47, the Ribbon shows the actions you can take on a site that is synchronized through SharePoint Workspace 2010.

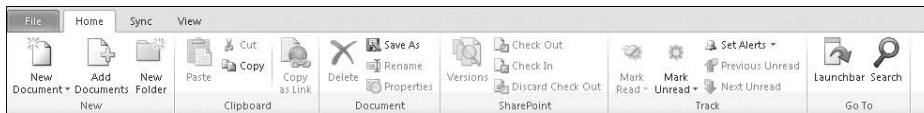


FIGURE 1-47 The SharePoint Workspace 2010 Ribbon

The Sync tab allows you to synchronize a site manually and provides you with the ability to redefine synchronization settings and check the status of the last synchronization operation. The View tab on the Ribbon is similar to the view settings in a Web browser—you can define and use different views on a list or library. In addition, you can use the View tab to open new workspace windows to view multiple sets of data at the same time, as shown in Figure 1-48.

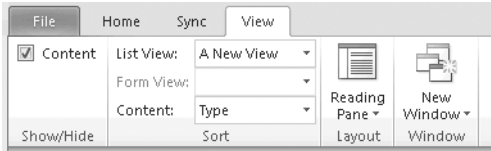


FIGURE 1-48 SharePoint Workspace 2010 View Ribbon

In general, the new SharePoint Workspace 2010 is a valuable tool that has been added to the SharePoint suite of products to enable users to interact and collaborate when they are out of the office and unable to access a SharePoint site directly.

Microsoft Word 2010

One of the new features in Microsoft Office 2010 is a Share option that allows you to publish a document to a SharePoint site without manually uploading the file first. After you create a document in Microsoft Word 2010, for example, you can click the File tab and then select Share to open the page shown in Figure 1-49. Then click the Save To SharePoint option and select a SharePoint site location, and the document will be uploaded automatically to the SharePoint site.

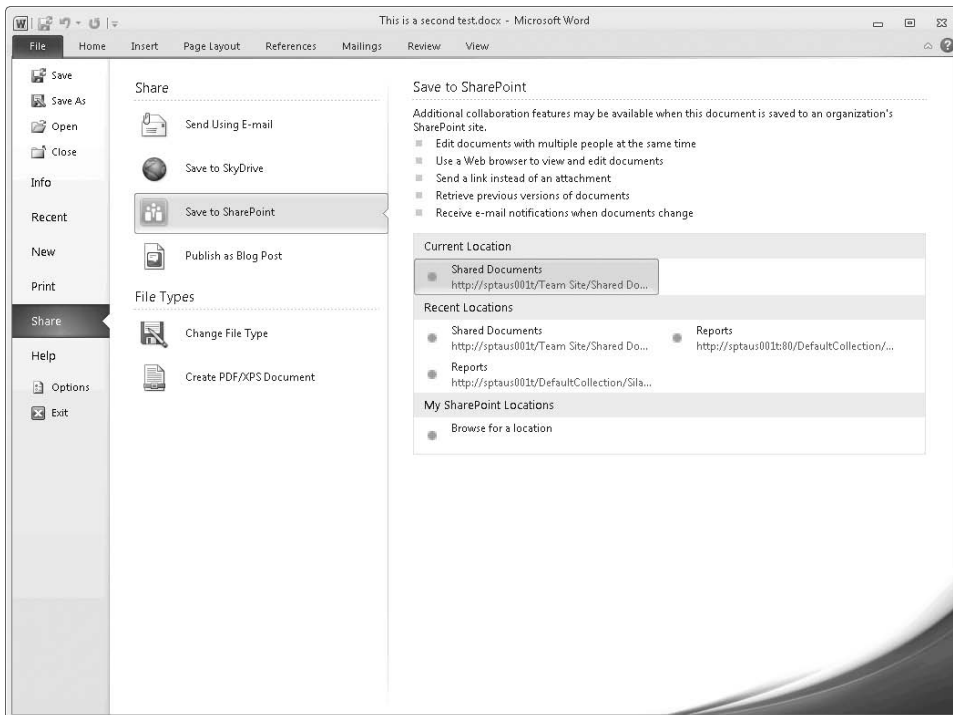


FIGURE 1-49 Uploading a document to a SharePoint site directly from Microsoft Word 2010

Also located under the File tab is the Info option, which displays SharePoint-related information about setting permissions, checking document properties, and version/check-out status. Figure 1-50 shows an example of the information that is displayed.

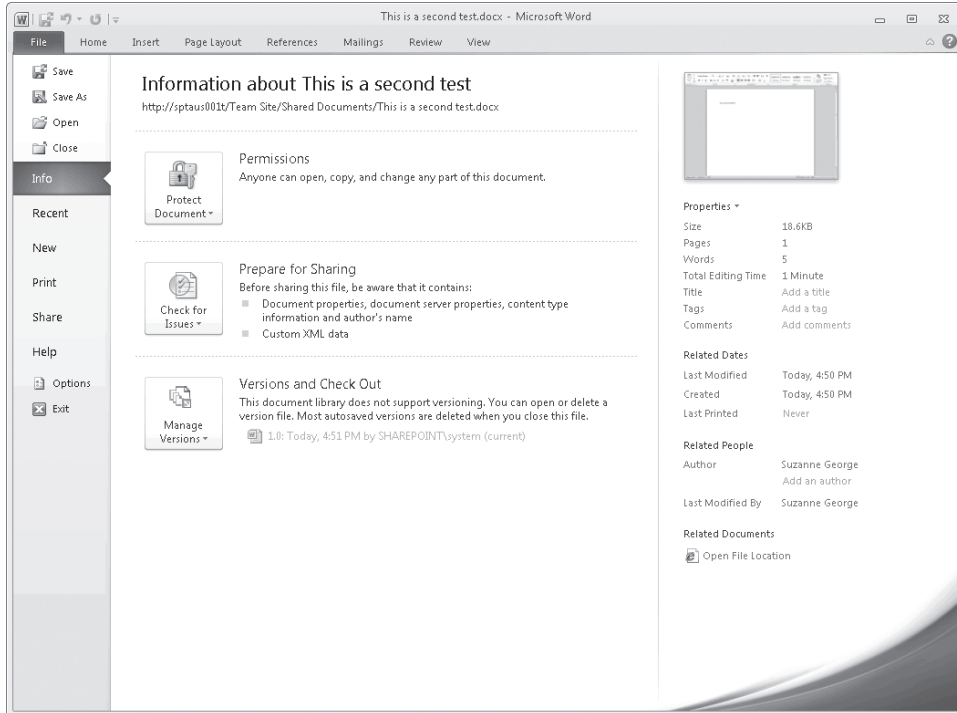


FIGURE 1-50 Microsoft Word file information displayed using the Info tab

Microsoft Word 2010 can also use SharePoint 2010 list data for mail merges and form information. The techniques for handling list data from SharePoint 2010 are similar for all Office products, such as Excel, and provide users with ubiquitous access to SharePoint information.

Support for Other Microsoft Office Products

With the release of SharePoint 2010 and Office 2010, the compatibility between applications is nearly seamless. Items such as the Ribbon allow a user to work, collaborate, and share information with colleagues, clients, and partners seamlessly and without concern for user locale, with only minimal training. Enhancements to products like Microsoft Excel 2010, for example, allow a user to process a very large list of line of business data from a remote system within seconds.

MORE INFO To see a list of features available in Microsoft Office 2010, go to <http://www.microsoft.com/office/2010/en/default.aspx>.

Microsoft Office 2010 and SharePoint 2010 allow users to work anywhere the need arises, collaborate better with SharePoint integration, and get work done faster than ever before.

Additional Information

This chapter has attempted to provide you with the basics of the collaboration and list functionality of SharePoint 2010. If you want to learn more, there are several resources available to administrators that provide in-depth information on how SharePoint 2010 works, including the following websites.

- SharePoint 2010 Technologies Page: <http://sharepoint2010.microsoft.com/Pages/default.aspx>
- System requirements and what's new in Microsoft SharePoint Foundation 2010: <http://technet.microsoft.com/en-us/sharepoint/ee263910.aspx#tab=1>
- System requirements and what's new in Microsoft SharePoint 2010 : <http://technet.microsoft.com/en-us/sharepoint/ee263917.aspx>
- Find a Microsoft partner to assist in a SharePoint Proof of Concept: <https://solutionfinder.microsoft.com>

Summary

SharePoint 2010 takes collaboration to a new level by offering organizations a mature, feature-rich platform to collaborate, manage information, and organize data in a smart, efficient manner. With the user interface and Office integration enhancements, SharePoint 2010 is setting the standard for collaboration systems. The remaining chapters in this book will help guide and advise you in making important administration decisions surrounding SharePoint 2010 to ensure a successful implementation and design for your unique enterprise or business.

In the next chapter, you'll turn your attention to the architecture of SharePoint 2010. Understanding how this product is built is essential for troubleshooting problems that arise, and you will learn about how SharePoint 2010 is built and how its components work together.

Using Windows PowerShell to Perform and Automate Farm Administrative Tasks

- Using Windows PowerShell: The Basics 195
- Working with SharePoint 2010 Management Shell 210
- Examples of SharePoint Administrative Tasks 234

Over the last couple of years, some SharePoint administrators have recognized the power of Windows PowerShell and have tried to use it, but with no built-in support, most administrators have found it too difficult. Microsoft SharePoint Foundation 2010 and Microsoft SharePoint 2010 now contain native support for Windows PowerShell with the SharePoint 2010 Management Shell. This chapter explores the SharePoint 2010 Management Shell and how to use Windows PowerShell to accomplish some basic administrative tasks. Chapter 12, “Using Windows PowerShell to Manage Search Services and FAST Search,” builds on the information in this chapter, using search administrative tasks and scenarios.

There is an enormous amount of information available about Windows PowerShell on the Internet and published in books. The aim of this chapter is not to teach you Windows PowerShell, but to show you how to become familiar with the new Windows PowerShell cmdlets, pronounced *command-lets*, introduced in SharePoint 2010. However, for those who are relatively new to Windows PowerShell, this chapter begins by reviewing the basics.

Using Windows PowerShell: The Basics

Traditionally, administrating a server on which Microsoft products are installed has involved learning a number of administrator tools, such as graphical interfaces based on the Microsoft Management Console (MMC), known as *MMC snap-ins*; many command-line utilities; and perhaps a Windows Management Instrumentation (WMI) provider or Component Object Model (COM) objects that administrators can interface with using

VBScript. Until the release of SharePoint 2010, SharePoint Products and Technologies were no different than other Microsoft products in the number of available administrator tools. Administrators needed to rely on the Central Administration website, the SharePoint Products Configuration Wizard, and the command-line tools stsadm and psconfig. No one tool could do everything.

Each Microsoft product also had its own community, which created or extended the built-in tools to make an administrator's life easier. The tools an administrator needed were available, but they were scattered among many sources. This is all changing now that every administrative task can potentially be scripted or automated using Windows PowerShell.

The concept for Windows PowerShell is based on a study commissioned by Microsoft in the early 2000s. Originally based on the POSIX shell as specified in IEEE 1003.2 and influenced by Perl and UNIX shell, Windows PowerShell is a command shell and scripting language that is far more powerful than using the Windows command prompt (cmd.exe). Administrators love command-line tools because they can be batched together to automate repetitive tasks or to ensure that a set of tasks are completed again exactly as they were the last time they were executed. Windows PowerShell cmdlets offer administrators these same benefits. You can still use the traditional tools, but Windows PowerShell adds flexibility and breadth that the traditional tools do not provide.

NOTE Community-led initiatives for previous versions of SharePoint include extensions to stsadm, published at <http://stsadm.codeplex.com/>; Windows PowerShell scripts for Windows SharePoint Services 3.0 and Microsoft Office SharePoint Server 2007 that can be found at <http://sharepointpsscripts.codeplex.com/>; and the CodePlex project "PowerShell SharePoint Provider" for Windows SharePoint Services 2.0 and Microsoft SharePoint Portal Server 2003, available at <http://www.codeplex.com/PSSharePoint>.

This section covers the following Windows PowerShell topics.

- What's new in the Windows PowerShell 2.0 release
- Architecture
- Installing Windows PowerShell
- Remote management

What's New in Windows PowerShell 2.0

Windows PowerShell 2.0 offers a number of new features and improvements from Windows PowerShell 1.0, including

- New cmdlets
- Remote Management and background jobs
- Windows PowerShell Integrated Scripting Environment (ISE)
- Windows PowerShell debugger

- Modules
- Advanced functions
- Transactions
- Steppable pipelines
- Events
- Script internationalization
- Online Help

As an IT professional, many of these new features will interest you. For example, you can remotely manage all the servers in your farm from your desktop, and when you become competent in Windows PowerShell, you will be able to create your own advanced functions using the Windows PowerShell scripting language that you can then use as cmdlets. In Windows PowerShell 1.0, the only way to do this was with a developer, using code and deploying files onto your farm.

MORE INFO For more information about what's new in Windows PowerShell 2.0, see <http://technet.microsoft.com/en-us/library/dd378784.aspx> and http://en.wikipedia.org/wiki/Windows_PowerShell. Other resources can be found by using your favorite Internet search engine and entering the keywords **What's New in Windows PowerShell 2.0** or the feature names listed previously.

Windows PowerShell Architecture

Why is Windows PowerShell so important? Windows PowerShell scripting language is object orientated, built on top of the .NET Framework—.NET Framework 3.5 for Windows PowerShell 2.0—and based on the C# (pronounced C-sharp) programming language. This allows Windows PowerShell to access the underlying object models, to pass objects and their values (properties) from one Windows PowerShell command to another. This means you have almost the power of a developer without having to write, compile, and deploy code. The disadvantage is that it can be all too easy to cause havoc in your installation.

When the Microsoft product teams consider the tools they want to provide to administrators, they build them first on Windows PowerShell. Hence, with the installation of SharePoint 2010, a large number of administrative tasks can be completed natively with Windows PowerShell cmdlets, with no need to install software from community-led initiatives. If any additional cmdlets are required, you can create them using Windows PowerShell scripts and the Advanced Functions feature in Windows PowerShell 2.0, and developers within your organization can create new cmdlets. Using Windows PowerShell to help administer SharePoint is the way of the future, so it's important to start learning to use the built-in cmdlets as soon as possible.

At first, using Windows PowerShell may seem daunting, but if you have used the command prompt (cmd.exe) or have created batch files, you will soon be comfortable with the Windows PowerShell console. In fact, you can even use the Windows PowerShell console instead of the command prompt, because all the commands you currently use, such as dir, cd, and ping, work equally well in the Windows PowerShell console as they do in the command prompt. You may need to enclose command-line parameters in quotation marks to use the commands from within Windows PowerShell, but that is about the only modification you need to make. So there is no need to use two different windows to complete your administrative tasks.

NOTE By default, the command prompt has a black background, but this can be changed. Right-click the title bar and then select Properties. A Properties dialog box appears with four tabs: Options, Font, Layout, and Colors. You can change the font, background color, window size, and more. See the sidebar titled “Command-line Shortcuts” later in this chapter for more information about similarities between the two command-line interfaces.

There are two ways of using Windows PowerShell: either through a command-line interface, known as the Windows PowerShell console (powershell.exe), or through the Windows PowerShell Interactive Scripting Environment (ISE) graphical interface (Powershell_ise.exe). This chapter concentrates on using the Windows PowerShell console.

MORE INFO Following are some resources for more information on Windows PowerShell.

- *Windows PowerShell 2.0 Administrator's Pocket Consultant*, by William R. Stanek (Microsoft Press, 2009)
- “Scripting with Windows PowerShell” on the Microsoft TechNet Script Center at <http://technet.microsoft.com/en-us/scriptcenter/dd742419.aspx>.
- An aggregation of Windows PowerShell bloggers at <http://pipes.yahoo.com/powershell/englishbloggers>.
- A short introduction to Windows PowerShell in a free eBook at <http://blogs.msdn.com/mapo/archive/2007/07/17/windows-powershell-free-ebook-at-microsoft.aspx>.
- *Mastering PowerShell*, by Dr. Tobias Weltner, a free eBook at <http://blogs.msdn.com/powershell/archive/2009/07/17/free-powershell-v1-book-from-the-makers-of-powershell-plus.aspx>.

Command-line Shortcuts

All the same shortcuts that you use at the command prompt work in the Windows PowerShell console.

- To pause the display temporarily when a command is writing output, press Ctrl+S, and then press Ctrl+S to resume or press Ctrl+C to terminate execution.
- By default, up to 50 commands are stored in a buffer. You can move through the buffer by using the up arrow and down arrow keys to scroll through the commands so you can easily execute the next or a previous command.
- The buffered commands can be displayed from a pop-up window by pressing F7. Select the command you want to execute using the arrow keys, press Enter or F9, type the number of the command you want to execute, and then press Enter.
- The buffered commands can be displayed from the command line by typing the first few characters of the command you want and then pressing F8. The command-line interface searches through the history buffer for the first command that begins with the characters you typed.
- Using the auto completion capabilities of the Windows PowerShell command shell, you never need to type the complete name of a command. Type in the first few characters of the command you want and press Tab to cycle through all commands and available file names and folders in alphabetic order, or click Shift+Tab to cycle through the commands in reverse order.
- To copy and paste text, right-click the command-line interface title, click Mark, highlight the text you want to copy, and then right-click the window title again to copy the selected text automatically into the Clipboard.
- Commands are case insensitive and can be batched together and placed in a text file. By calling that file in the command-line window, the commands in the file are executed. These text files have an extension of .bat, .cmd, .vbs, or in the case of Windows PowerShell, .ps1. Commands in these files can be executed outside the command window by calling the file from the Run line or by double-clicking the file in Windows Explorer.
- Arguments can be sent to the command file being called. The arguments passed to your command file are stored in memory and accessed using variables. You can also create variables to store values needed as you complete a task. In Windows PowerShell, you reference variables by preceding the variable name by a dollar sign (\$).
- Output from a command can be directed to a file named to the right of >. This will overwrite the file if it already exists. To redirect the output but append it to the file, use >>. Error messages that result from running a command can be redirected to a file named to the right of 2>, or they can

be appended to a file by using `2>>` or sent to the same destination as the standard output by using `2>&1`.

- You can execute many commands on the same line. In Windows PowerShell, you separate commands using the semicolon character (;).
- Redirect the output of one command as input to another command by separating both command with a '|', known as a *pipe*. For example, `dir | sort | more` will display a sorted list of files in the current directory one page at a time. The pipe is the symbol located above the backslash on most keyboards.
- Aliases and shortcuts can be substituted for commonly used commands. For example, the `ForEach-Object` cmdlet can be replaced with `ForEach` or even with the percentage character, `%`. The question mark character, `?`, can be used in place of `Where` or `Where-Object`. You can also use `Get-Alias` to return a list of aliases. Note that the use of aliases in scripts can make them difficult to understand.

Examples of these shortcuts can be found throughout this book.

Installing Windows PowerShell

Computers running Windows 7 or Windows 2008 R2 or later include Windows PowerShell 2.0 and Windows Remote Management (WinRM) 2.0. If you want to manage computers using earlier operating systems locally or remotely using Windows PowerShell, you will need to install both Windows PowerShell 2.0 and WinRM 2.0.

Installing Windows PowerShell 2.0 and WinRM 2.0

A copy of Windows PowerShell 2.0, together with WinRM 2.0, can be downloaded from <http://support.microsoft.com/default.aspx/kb/968929>. Ensure that you obtain the correct version; for example, if you install SharePoint on the 64-bit version of Windows Server 2008, you need to download and install the Windows 2008 x64 version of Windows PowerShell 2.0 and WinRM 2.0. After they are installed, you will find the executable for the Windows PowerShell ISE in the folder `%SystemRoot%\System32\WindowsPowerShell\v1.0`.

NOTE If you going to be a heavy user of ISE, consider placing a shortcut for this program on your taskbar. On the Start menu, right-click `Powershell_ise.exe` and then select either **Pin To Taskbar** or **Pin To Start Menu**.

To use Windows PowerShell ISE on Windows 2008 R2, you need to add the Windows PowerShell ISE feature. This feature can be added using Windows PowerShell commands or the Server Manager. After you have installed SharePoint 2010 or SharePoint Foundation 2010, perform the following steps.

- Using Windows PowerShell, enter the following commands.


```
Import-Module Servermanager; Add-WindowsFeature "PowerShell-ISE"
```
- Using Server Manager, complete the following procedure.
 1. Start the Server Manager, click Features, and then select Add Features.
 2. In the middle pane of the Add Features Wizard, select the check box for Windows PowerShell Integrated Scripting Environment (ISE) as shown in Figure 5-1 and then click Next.

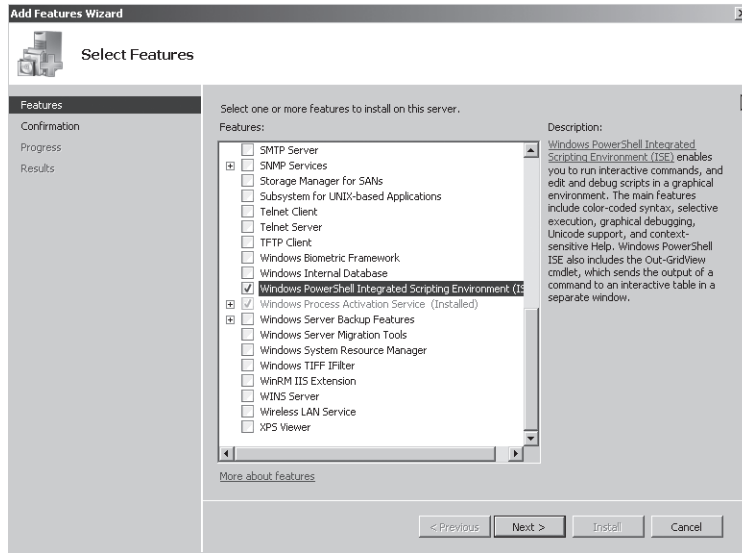


FIGURE 5-1 Install the Windows PowerShell Integrated Scripting Environment (ISE) feature

3. On the Confirmation Installation Selection screen, click Install.
4. On the Installation Results screen, click Close.

NOTE You may need to restart your server after the installation completes.

Working Within Your Execution Policy

Windows PowerShell is a powerful tool, and as with any other scripting language, it is all too easy to borrow someone else's code or download snippets from the Internet. Also, because the code files are just text files, it is easy for anyone to modify them and inject malicious code. Therefore, Windows PowerShell has a built-in security feature called *execution policy* that you can set on a per-user basis, and these settings are saved in the registry. To view your execution policy, type the following command in the Microsoft SharePoint 2010 Management Shell.

```
Get-ExecutionPolicy
```


If you are running the Management Shell on a SharePoint server, it is likely that the output from such a command is RemoteSigned, which means that you can run any commands interactively and you can use any scripts that are stored on the computer where you are logged in. However, if you want to run configuration files or scripts from remote sources, they need to be signed. You can change the execution policy if you start the Windows PowerShell console as an administrator and use the Set-ExecutionPolicy cmdlet, or you can change the registry key as follows.

```
HKLM\Software\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell
```



SECURITY ALERT The registry key is useful in conjunction with an Active Directory Group Policy. Manual modifications to the registry should be avoided, especially when Microsoft provides a cmdlet to configure Windows PowerShell execution policy.

To sign scripts, use the Set-AuthenticodeSignature cmdlet, which can be used to add an Authenticode signature to a Windows PowerShell script or other file.

In an organization that has multiple environments, such as development, system integration, user acceptance test (UAT), pre-production, and production, consider the need to sign code on your production and UAT farms at minimum. You should also review the execution policy settings for these farms. You can set the execution policy in a Group Policy Object that targets either users or computers so that it cannot be overridden when someone logs on to your SharePoint servers.

MORE INFO For more information on Windows PowerShell execution policies and signing scripts, see the TechNet article “Heading Off Malicious Code” at <http://technet.microsoft.com/en-us/magazine/2008.01.powershell.aspx>, or type **Get-Help about_Execution_Policies** at a Windows PowerShell command-line interface. There is also a blog post that explains the process in detail. It is “ALLSigned: Signing Your PowerShell Scripts” and can be found at <http://sharepoint.microsoft.com/blogs/zach/Lists/Posts/Post.aspx?ID=53>.

NOTE If you are using Windows 7 as your desktop and you want to remotely manage Windows Server 2008 R2 for your SharePoint servers, you need the Remote Server Administration Tools (RSAT), which is required for the new Group Policy features and supports Windows PowerShell. RSAT is available from the Microsoft Download Center at <http://www.microsoft.com/downloads>. For more information, go to <http://technet.microsoft.com/en-us/library/dd367853.aspx> and <http://trycatch.be/blogs/roggenk/archive/2009/06/08/installing-windows-7-rsat-unattended.aspx>.

Managing Systems Remotely with WinRM

Windows PowerShell 2.0 introduces a new capability to manage your systems remotely from your desktop by using either WinRM or Internet Information Server (IIS). WinRM is often the mechanism used by administrators and the subject of this section. Remote Management involves not just the ability to run Windows PowerShell locally on your machine using the few commands that allow you to specify a computer name as an optional parameter; it also includes methods known as fan-in and fan-out remoting and background jobs.

- **Fan-in remoting** This allows many administrators to connect to an instance of Windows PowerShell running on the same remote servers—this is not supported out of the box in SharePoint 2010.
- **Fan-out remoting** This allows you to send a single Windows PowerShell command to run multiple remote instances of Windows PowerShell in parallel, and the results of those commands will be returned to your desktop. You would use this if you need to complete the same task on multiple servers. You no longer need to establish a Remote Desktop connection to each server in turn and then execute the commands locally on that server. You can create a set of Windows PowerShell commands and pass the server names to those commands, which then completes the same commands sequentially for each server whose name you provide.
- **Background jobs** Windows PowerShell 2.0 supports both local and remote background jobs. These are commands that execute asynchronously in the background with no interaction. When you execute a command in the background, the command prompt is returned immediately so that you can continue to execute other commands.

Configuring for WinRM

The WinRM Windows service must be started and configured for remoting on both your local computer and the server on which you want to remotely run commands. To find out if your server is running WinRM, type

```
Get-service winrm
```

To check if it is running on a remote server, type

```
Get-service winrm -computername $server_name
```

On your SharePoint server, this service should be running; however, if your local computer is running Windows 7 (or Windows Vista with Windows PowerShell 2.0 and WinRM 2.0 installed), you may need to start this service and enable remoting. You can do this by typing one command, `Enable-PSRemoting`, which executes two additional commands, `Set-WSManQuickConfig` and `Start-Service WinRM`.

IMPORTANT If one of the network connection types is `Public`, the `Set-WSManQuickConfig` command will raise an `Invalid Operation Exception`. You will need to change the network connections to the network type of either `Domain` or `Private` to proceed.

The output to the Enable-PSRemoting command will look similar to the following example.

```
WinRM Quick Configuration
Running command "Set-WSManQuickConfig" to enable this machine for remote management
through WinRM service.
This includes:
  1. Starting or restarting (if already started) the WinRM service
  2. Setting the WinRM service type to auto start
  3. Creating a listener to accept requests on any IP address
  4. Enabling firewall exception for WS-Management traffic (for http only).

Do you want to continue?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default
is "Y"): y
WinRM has been updated to receive requests.
WinRM service type changed successfully.
WinRM service started.
Configured LocalAccountTokenFilterPolicy to grant administrative rights remotely to
local users.
WinRM has been updated for remote management.
Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
WinRM firewall exception enabled.
```

On a computer running a 64-bit version of Windows, you may see an additional confirma-
tion message.

```
Are you sure you want to perform this action?
Performing operation "Registering session configuration" on Target "Session
configuration "Microsoft.PowerShell32" is not found. Running command
"Register-PSSessionConfiguration Microsoft.PowerShell32 -processorarchitecture x86
-force" to create "Microsoft.PowerShell32" session configuration. This will restart
WinRM service."
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

To determine how WinRM is configured, the authentication method, and the port numbers
it will access, use the following command.

```
winrm get winrm/config/service
```

This should result in an output similar to the following.

```
Service
RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)S:P(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD)
MaxConcurrentOperations = 4294967295
MaxConcurrentOperationsPerUser = 15
```

```

EnumerationTimeoutms = 60000
MaxConnections = 25
MaxPacketRetrievalTimeSeconds = 120
AllowUnencrypted = false
Auth
    Basic = false
    Kerberos = true
    Negotiate = true
    Certificate = false
    CredSSP = false
    CbtHardeningLevel = Relaxed
DefaultPorts
    HTTP = 5985
    HTTPS = 5986
IPv4Filter = *
IPv6Filter = *
EnableCompatibilityHttpListener = false
EnableCompatibilityHttpsListener = false
CertificateThumbprint

```

You can run winrm and winrm-related commands, such as the New-PSSession command, from either the Windows PowerShell console or the SharePoint 2010 Management Shell. However, these must be run as an administrator. To do this, right-click the Windows PowerShell console or the SharePoint 2010 Management Shell and then select Run As Administrator. You must be a member of the Administrators group on the remote machine or be able to provide administrator credentials to do this.

NOTE When your local computer and the remote computer are in different non-trusted domains or in a workgroup, additional steps are required to configure remoting. The workgroup scenario is likely to occur when SharePoint Foundation is installed for external collaboration or in a small business environment rather than in a typical SharePoint Server installation. For information on how to configure remoting between two workgroup machines, see the blog post at <http://blogs.msdn.com/wmi/archive/2009/07/24/powershell-remoting-between-two-workgroup-machines.aspx>.

Meeting Requirements for SharePoint PowerShell Remoting

To use WinRM to execute SharePoint PowerShell cmdlets on a remote machine, there are a number of additional requirements.

1. You must have access to a userid with the correct rights to execute those commands. These requirements are applicable whether you are remoting into a server or locally executing SharePoint cmdlets. The userid must meet the following criteria.

- Must be a member of the WSS_ADM_WGP local security group on the machine executing the commands.
- Must be a member of the SharePoint_Shell_Access SQL Server role on the configuration database.
- Must have access to the content database that you want to manipulate.

You can use the SPShellAdmin cmdlets to help you manage these requirements. When you run the Add-SPShellAdmin cmdlet to add a userid to the SharePoint_Shell_Admin role, you must be mapped to Security_Admin role on the SQL instance and the db_owner role to the relevant databases. You need to run the Add-SPShellAdmin cmdlet for each content database that you want the userid to access.

2. Designate one server in your farm as the server you will always use for remoting. On that server, increase the WinRM configuration option, MaxMemoryPerShellDB, to a large value, 512 or 1024. This configuration option limits the amount of memory a single remote process can use. Increasing the value will allow long-running commands to run without throwing OutOfMemory Exception errors. The command to display the values of the configuration options and sample output is shown here.

```
winrm get winrm/config/winrs
```

```
Winrs
  AllowRemoteShellAccess = true
  IdleTimeout = 180000
  MaxConcurrentUsers = 5
  MaxShellRunTime = 2147483647
  MaxProcessesPerShell = 15
  MaxMemoryPerShellMB = 150
  MaxShellsPerUser = 5
```

To modify the MaxMemoryPerShellDB setting, type the following Windows PowerShell command in the SharePoint 2010 Management Shell.

```
Set-Item WSMAN:\localhost\Shell\MaxMemoryPerShellMB 1024
```

3. You must use Credential Security Provider (CredSSP) authentication. This authentication mechanism, introduced with Windows Vista, allows an application to delegate the credentials from one machine to another, a process called *double hopping*. This will enable you to provide the credentials that allow the SharePoint PowerShell cmdlets to talk to the Microsoft SQL Server that is hosting your SharePoint databases. If you do not use CredSSP and you try to remotely execute a SharePoint cmdlet, you are likely to receive a message that indicates the farm is unavailable. In the output from the Winrm get winrm/config/service command shown previously, notice that CredSSP is equal to False. There are several ways you can modify WinRM configuration options, including Group Policy, login scripts, and the command prompt. To enable CredSSP authentication through command-line interfaces, complete the following tasks.

- a. On the computer from which you are remoting, such as your Windows 7 desktop, type the following command, where Web1 is the SharePoint server. Type **Y** when prompted to confirm that you want to enable CredSSP authentication.

```
Enable-WManCredSSP -role client -DelegateComputer Web1
```

- b. On the SharePoint server where you are remoting to, type

```
Enable-WManCredSSP -role server
```

MORE INFO For more information on multi-hop support in WinRM, see [http://msdn.microsoft.com/en-us/library/ee309365\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ee309365(VS.85).aspx).

Using WinRM Remoting

The commands you need to start a remote session to remotely manage a SharePoint farm or one specific SharePoint server are similar to those in the following example.

```
$cred = Get-Credential contoso\spfarm
$sess = New-PSSession Web1 -Authentication CredSSP -Credential $cred
Invoke-Command -Session $sess -ScriptBlock `
    {ADD-PSSnapin Microsoft.SharePoint.PowerShell;}
```



SECURITY ALERT You can limit the Windows PowerShell commands that a user can access during a remote session by using the `-ConfigurationName` parameter with the `New-PSSession` cmdlet or by using the `Proxy` cmdlet. For more information, see the blog post “Extending and/or Modifying Command with Proxies” at <http://blogs.msdn.com/powershell/archive/2009/01/04/extending-and-or-modifying-commands-with-proxies.aspx>.

When you enter the first command, a dialog box appears where you must type the password. This password and the user name are stored in the variable `$Cred`, which is used in the second command, `New-PSSession`, which establishes a persistent session with the remote server. Creating a persistent session is not necessary when using Windows PowerShell remoting, but if you do not establish a persistent connection, whenever you invoke a remote command that includes a SharePoint cmdlet, each `Invoke-Command` will also need to include the `PSSnapin` cmdlet. The third command, `Invoke-Command PSSnapin`, allows you to run SharePoint PowerShell cmdlets on the remote server.

Reference to the persistent session is stored in the variable `$sess`. You can direct any commands you want to that remote server by using this variable. In the following example, the cmdlet `Get-SPServiceInstance` is typed within the script block—the area between the curly brackets (`{ }`). The output from the `Get-SPServiceInstance` cmdlet returns the status of the services within a SharePoint farm.

```
Invoke-Command -Session $s -ScriptBlock {get-SPServiceInstance}
```

```
WARNING: column "PSComputerName" does not fit into the display and was removed.
```

TypeName	Status	Id
-----	-----	--
Business Data Connectivity	Online	3ad4d004-9ae3-4810-94bf-76cc43c9d507
Microsoft SharePoint Foundati...	Online	81cc6474-fd6f-42ad-a932-25d67cff8cc1
Microsoft SharePoint Foundati...	Disabled	cb3b976c-b451-4abe-b808-0c6d191ccd3d
Microsoft SharePoint Foundati...	Online	5bac9c93-483a-4901-ae46-f6f7ae0a12a8
Central Administration	Online	0c24457f-b58e-4daa-895e-4acd35ef7543
Microsoft SharePoint Foundati...	Online	155570fa-0d4c-495d-9f31-5250e83bdd75
Microsoft SharePoint Foundati...	Disabled	c25ae170-c07d-48db-97b3-73b0d9b47e94
SharePoint Foundation Help Se...	Online	8af8b8b1-92cb-48d2-8864-20a7e191d7c2

BEST PRACTICE You can type multiple commands in a script block, either on separate lines or on the same line separated by a semicolon (;). If only one command is entered in the script block, the semicolon is optional, but it is good practice to include it. You can also use local variables to generate a remote script block as in the following example.

```
$url = "http://contoso.com"  
$myscript = "get-spsite $url"  
$sb = [scriptblock]::Create($myscript)  
Invoke-Command $sess -ScriptBlock $sb
```

You can use the Windows PowerShell `Invoke-Command` cmdlet to communicate with many sessions, either on the same server or on different servers, from your desktop. This cmdlet enables you to start multiple administrative tasks at the same time that are then run in parallel. However, if these tasks are long running, you will not get control back until the command on the remote machine finishes. This is known as *running the command interactively*. To run asynchronously as a background job so that the Windows PowerShell prompt returns immediately and you can enter other commands, append the `-AsJob` parameter to the `Invoke-Command`, or alternatively, use the `Start-Job` cmdlet. Using this technique can reduce the amount of time to complete your administrative tasks, compared to running them sequentially. An example of how this can save you a considerable amount of time is during the upgrade process, when each session runs a database-attach-upgrade, and the time taken to complete all database upgrades is limited only by the resources of your SQL Server.

If you have many short-running tasks to complete on a specific server and don't want to keep typing `Invoke-Command`, use one of the following approaches.

- You can enter an interactive session with that server by typing **Enter-PSSession \$sess**. The command prompt will change from PS C:\Users*<userid>*, where *<userid>* is the name of the current user, to [*<remoteservername>*]: PS C:\Users*<remotecredentials>*\Documents, where *<remotecredentials>* is the userid you used for the CredSSP authentication. The command prompt reminds you that you are now submitting commands to the remote server. To return to interactive mode on your local computer, type **Exit-PSSession** or **exit**.
- Use the Import-PSSession remoting cmdlet, also known as *implicit remoting*, to bring commands from the remote session into the local Windows PowerShell session. An additional advantage of the Import-PSSession approach is that you can interact with your local file system and talk to SharePoint as if you were logged on locally to the SharePoint server. By default, the Import-PSSession cmdlet imports all commands except commands that have the same name as commands in your current session. You can also import a subset of commands and prefix them with a word you specify, so that it is obvious to you which commands are local and which are remote. In the following example, the word *Remote* is added as a prefix to all SPSite cmdlets.

```
Import-PSSession -session $sess -CommandName *-spsite -Prefix Remote
```

ModuleType	Name	ExportedCommands
Script	tmp_1833f7bc-b269-4229...	{Set-SPSite, Backup-SPSite, Get-SPSite, ...}

In this example, you can use all SPSite cmdlets as if they were local commands; for example, by typing **Get-RemoteSPSite** you can return all site collections that match the given criteria. These imported commands are stored in a temporary module that is deleted at the end of the session. To create a persistent module that you can use in future sessions, use the Export-PSSession cmdlet. The imported commands still run in the remote session from which they were imported and therefore may take longer to run than local commands.

When you have completed all of your tasks, you can delete the persistent connection between your local machine and the remote machine by typing **Remove-PSSession \$sess**.

MORE INFO For more information about Windows PowerShell remoting, see Chapter 4, "Using Sessions, Jobs, and Remoting" in the *Windows PowerShell 2.0, Administrator's Pocket Consultant*, by William Stanek (Microsoft Press, 2009), and the TechNet Magazine article "A Sneak Peek at Remote Management in Version 2.0" at <http://technet.microsoft.com/en-us/magazine/2008.08.windowspowershell.aspx>. Also take a look at the blog post "A Few Good WS-MAN (WinRM) Commands" at <http://blogs.technet.com/otto/archive/2007/02/09/sample-vista-ws-man-winrm-commands.aspx>.

Working with the SharePoint 2010 Management Shell

To open the SharePoint 2010 Management Shell, click Start, and then select Programs, Microsoft SharePoint 2010 Products, and finally, SharePoint 2010 Management Shell, as shown in Figure 5-2.



FIGURE 5-2 Accessing the SharePoint 2010 Management Shell

The SharePoint 2010 Management Shell is a custom console; it is not the same console that opens if you open the default Windows PowerShell console, a shortcut to which is usually placed on the taskbar. A review of the properties of the SharePoint 2010 Management Shell shortcut exposes the command that is executed.

```
C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe -NoExit " &
' C:\Program Files\Common Files\Microsoft Shared\Web Server
Extensions\14\CONFIG\POWERSHELL\Registration\sharepoint.ps1 ' "
```

The shortcut points to the file `Sharepoint.ps1`. This is an example of a Windows PowerShell profile, which is identical to any other Windows PowerShell script file and is used to store frequently used elements that need to be automatically loaded at the start of a Windows PowerShell session. Hence, just like normal scripting files, profiles can contain functions, aliases, and variables, and they can load any Windows PowerShell extensions, snap-ins, or modules you might need. A Windows PowerShell snap-in (PSSnapin) is a .NET program that is compiled into DLL files and contains new cmdlets, functions, and/or providers. Windows PowerShell comes with a number of snap-ins, including `Microsoft.PowerShell.Core`, `Microsoft.PowerShell.Host`, and `Microsoft.PowerShell.WSMan.Management`, as well as modules such as

ActiveDirectory, FailoverClusters, and WebAdministration. As other products are installed on your computer, additional extensions will become available.

The main aim of the SharePoint profile file is to load the Windows PowerShell for SharePoint snap-in so you can then use the new cmdlets specific to SharePoint. The SharePoint profile file contains the following code, plus a signature.

```
$ver = $host | select version
if ($ver.Version.Major -gt 1) {$Host.Runspace.ThreadOptions = "ReuseThread"}
Add-PsSnapin Microsoft.SharePoint.PowerShell
Set-Location $home
```

This code obtains the version of Windows PowerShell, checks that it is greater than 1, and then if that is true, it sets the threading model so that the first thread will be reused, loads the SharePoint PowerShell snap-in, and then changes directory to the home folder, such as `C:\users\<userid>`, where *userid* is the person who is currently logged in. For more information on SharePoint, Windows PowerShell, and threading, see the sidebar titled “Memory Considerations When Using Windows PowerShell” later in this chapter.

The SharePoint PowerShell snap-in is not the only snap-in loaded into the SharePoint 2010 Management Shell. Therefore, not only can you use the SharePoint cmdlets, you also can use other cmdlets from the snap-ins that are loaded. To see all the snap-ins and the order in which they were loaded, type **Get-PSSnapin**. Your output should look similar to the following example.

```
Name          : Microsoft.PowerShell.Diagnostics
PSVersion     : 2.0
Description   : This Windows PowerShell snap-in contains Windows Eventing and
                Performance Counter cmdlets.

Name          : Microsoft.WSMan.Management
PSVersion     : 2.0
Description   : This Windows PowerShell snap-in contains cmdlets (such as
                Get-WSManInstance and Set-WSManInstance) that are used by the
                Windows PowerShell host to manage WSMan operations.

Name          : Microsoft.PowerShell.Core
PSVersion     : 2.0
Description   : This Windows PowerShell snap-in contains cmdlets used to manage
                components of WindowsPowerShell.

Name          : Microsoft.PowerShell.Utility
PSVersion     : 2.0
Description   : This Windows PowerShell snap-in contains utility Cmdlets used to
                manipulate data.
```

```
Name      : Microsoft.PowerShell.Host
PSVersion : 2.0
Description : This Windows PowerShell snap-in contains cmdlets (such as Start-
              Transcript and Stop-Transcript) that are provided for use with the
              Windows PowerShell console host.

Name      : Microsoft.PowerShell.Management
PSVersion : 2.0
Description : This Windows PowerShell snap-in contains management cmdlets used to
              manage Windows components.

Name      : Microsoft.PowerShell.Security
PSVersion : 2.0
Description : This Windows PowerShell snap-in contains cmdlets to manage Windows
              PowerShell security.

Name      : Microsoft.SharePoint.PowerShell
PSVersion : 1.0
Description : Register all administration Cmdlets for Microsoft SharePoint Server.
```

NOTE If you develop your own scripts that can be called from the command prompt or by double-clicking the file in Windows Explorer, and those scripts contain Windows PowerShell for SharePoint commands, make sure that you include the command to set the threading model and load the Windows PowerShell for SharePoint snap-in in your scripts. When calling a script from within a Windows PowerShell console, you can use the `#requires` tag at the top of your script.



REAL WORLD Customizing Your Console

It is common practice to modify the profile file on the designated administrator machine. The profile could be altered so that it

- Contains all the snap-ins or modules used by all IT professionals in an organization or department.
- Tracks all Windows PowerShell commands executed.
- Changes the prompt to indicate the server name. In an organization with a multitude of servers, it is all too easy to start an administrative task on the wrong server, so anything that can help an administrator identify the server he is working on will be helpful, such as configuring the window title to list the name of the server.

An example of a custom profile could contain the following code.

```
<# *****
Copyright (c)2010, Contoso, All Rights Reserved
*****
Contoso PowerShell Profile - ContosoProfile.ps1
Author: Peter Connelly
Purpose: This profile sets the PowerShell window size, font and title
and loads the SharePoint snap-in
History: Version 1.0 01/02/2010 First version
#>
# Track all Windows PowerShell commands
$profileName = $MyInvocation.MyCommand.Name
$profilePath = $MyInvocation.MyCommand.Path
$transcriptFile = "C:\Contoso\Logs\Powershell_{$profileName}.log"
Start-Transcript $transcriptFile -append -force
Write-Output "Starting profile: $profilePath"

# Command from SharePoint Server 2010 profile file - SharePoint.ps1
$ver = $host | select version
if ($ver.Version.Major -gt 1) {$Host.Runspace.ThreadOptions = "ReuseThread"}
Add-PSnapin Microsoft.SharePoint.PowerShell
Set-Location $home

# Check that this is a command-line interface and not the ISE
if ($host.name -eq "ConsoleHost")
{
    $width = 80
    $sizeWindow = new-object System.Management.Automation.Host.Size $width,40
    $sizeBuffer = new-object System.Management.Automation.Host.Size $width,9999
    <# Check to adhere to the following rules:
    The buffer width can't be resized to be narrower than the window's current
    width plus the window's width can't be resized to be wider than the
    buffer's current width. #>
    $s = $Host.UI.RawUI
    if ($s.WindowSize.width -gt $width)
    {
        $s.WindowSize = $sizeWindow
        $s.BufferSize = $sizeBuffer
    } else {
        $s.BufferSize = $sizeBuffer
        $s.WindowSize = $sizeWindow
    }
}
# Set foreground, background color and window title
$s.ForegroundColor = "Yellow";$s.BackgroundColor = "DarkBlue";
$s.WindowTitle = "$env:computername"
# *****
# End of ContosoProfile.ps1
# *****
```

Because the profile script is automatically executed every time you run the SharePoint 2010 Management Shell, it should be signed. More information about customizing your console can be found at <http://technet.microsoft.com/en-us/library/ee156814.aspx>.

Understanding cmdlets

Windows PowerShell cmdlets are utilities that are built on top of application interfaces, and in the case of SharePoint 2010, the SharePoint Object Module, and SharePoint Web Services, that allow administrators to complete a number of tasks.

In general, cmdlets use a verb-noun pair. The noun specifies the object you want information about or that you want to manipulate, and the verb states what you want to do with that object. The verbs and nouns are always separated by a hyphen with no spaces, and SharePoint cmdlet verbs have a prefix of SP. For example, if you want to get (the verb) information about all the content databases (noun) in your farm, you would use the following SharePoint cmdlet, which would give you the output shown.

```
PS C:\Users\Peter> Get-SPContentDatabase
```

```
Id           : a1d5c96c-a41a-43b3-bc5d-3f8a93b26046
Name        : WSS_Content_a2fde53006e04bf5aae434ffd3c8a19c
WebApplication : SPWebApplication Name=SharePoint - 80
Server      : SQL
CurrentSiteCount : 1

Id           : 363b11a3-6947-42f6-9df4-665eeff59c83
Name        : SPF_TeamsDB
WebApplication : SPWebApplication Name=SPF_Teams
Server      : SQL
CurrentSiteCount : 1
```

Windows PowerShell also provides a few cmdlets that help you to work with the other cmdlets. Two examples of these are `Get-Command`, which can be shorted to `gcm`, and `Get-Help`, which can be shorted to `help`. (Shortened names of cmdlets are called *aliases*.) `Get-Command` allows you to find cmdlets, and then `Get-Help` can provide you with basic information about a cmdlet after it is retrieved. (You'll see how to use the `Get-Help` cmdlet in the section titled "Getting Help" later in this chapter.) In the following example, the `Get-Command` finds all the cmdlets associated with SharePoint.

```
PS C:\Users\Peter>Get-Command -PSSnapin "Microsoft.SharePoint.PowerShell" |
>>sort noun, verb |Format-Wide -Column 3
>><ENTER>
```

NOTE When you type a one-line Windows PowerShell script, or you have not provided all the required parameters, Windows PowerShell either prompts you for the parameter or starts a new line with >> so that you can enter more commands. When you have finished typing, press Enter for Windows PowerShell to execute the code. Because of page size restrictions, many of the commands in this chapter are shown on multiple lines when they could be typed on one line, so always remember to press Enter only to execute the scripts.

The set of commands shown previously takes the output from the Get-Command cmdlet and pipes it to the sort cmdlet. The sort cmdlet sorts the data in noun-then-verb order so that all SharePoint cmdlets that manipulate the same object are listed together. The cmdlets that manipulate the same object are then sorted in verb order. To reduce the amount of scrolling necessary in the SharePoint 2010 Management Shell, the sorted data is piped to the format-wide cmdlet. Such a combination of Windows PowerShell cmdlets is quite common. The output from the command set would look similar to the following sample.

```
Start-SPAdminJob           Get-SPA1ternateURL       New-SPA1ternateURL
Remove-SPA1ternateURL     Set-SPA1ternateURL       Install-SPApplicationC...
Start-SPAssignment       Stop-SPAssignment        Get-SPAuthenticationPr...
New-SPAuthenticationPro... Get-SPBackupHistory      Move-SPBlobStorageLoca...
Get-SPBrowserCustomerEx... Set-SPBrowserCustomerEx... Copy-SPBusinessDataCat...
....(not all output shown)
Set-SPWebApplicationHtt... Get-SPWebPartPack        Install-SPWebPartPack
Uninstall-SPWebPartPack  Get-SPWebTemplate        Install-SPWebTemplate
Set-SPWebTemplate        Uninstall-SPWebTemplate  Get-SPWorkflowConfig
Set-SPWorkflowConfig
```

NOTE There is another cmdlet similar to the Get-Command that helps you work with cmdlets and that you might find useful—especially if you create your own commands or scripts. It is the Measure-Command cmdlet. This command allows you to measure the time it takes to run cmdlets or scripts. You may also find the Trace-Command cmdlet useful for debugging short script blocks.

Working with Objects

There are a considerable number of objects available in SharePoint, and as you might expect, there are more available in SharePoint Server than in SharePoint Foundation. By reading this book, you might have determined the relationship between these objects. For example, while a SharePoint farm can contain many Web applications, each SharePoint farm has one configuration database. Many of the SharePoint cmdlets use names that you can easily identify to determine which objects they will manipulate. To illustrate, the SPFarm cmdlets allow you

to retrieve information about and back up or restore a SharePoint farm. For example, the `SPFarmConfig` cmdlet lets you retrieve and set properties for the farm.

However, there are a few cmdlet names that could confuse you. For example, the `SPSite` cmdlet allows you to manipulate site collections, and Chapter 2, “Understanding the Architecture of SharePoint 2010,” explains that you can’t create a site collection unless you have a Web application, which means using the `SPWebApplication` object. When a site collection is created, you choose a site template to create a site. In the SharePoint Object Model, this site is known as a *web* and not a site; hence, the object is named `SPWeb`. This can be quite confusing because the term “website” is often shortened to “site,” and in the SharePoint Object Model, “site collection” is also shortened to “site.” Because this terminology has been used through many versions of SharePoint and is used consistently by developers, it is easier for IT professionals to adopt this usage.

- A *web* is a website and can be manipulated using the `SPWeb` object.
- A *site collection* contains a collection of webs and can be manipulated using the `SPSite` object.

Figure 5-3 displays the most common objects and their relationships, but there are others, such as `SPSiteCollection`, which is a collection of `SPSites`.

The current selection of built-in cmdlets does not provide full coverage of all SharePoint objects, but you can write new cmdlets to manipulate these objects. To determine the objects associated with the SharePoint PowerShell cmdlets and the number of cmdlets per each object, use the commands in the following example, where a portion of the output is shown.

```
PS C:\Users\peter> Get-Command -PSSnapin "Microsoft.SharePoint.PowerShell" |
>>sort noun, verb | group-object -property noun
>><ENTER>
```

```
Count Name                               Group
-----
1 SPAdminJob                             {Start-SPAdminJob}
4 SPAlternateURL                          {Get-SPAlternateURL, New-SPAlternateURL, Rem...
1 SPApplicationContent                    {Install-SPApplicationContent}
2 SPAssignment                             {Start-SPAssignment, Stop-SPAssignment}
...(not all output shown)
6 SPWeb                                   {Export-SPWeb, Get-SPWeb, Import-SPWeb, New-...
4 SPWebApplication                        {Get-SPWebApplication, New-SPWebApplication,...
1 SPWebApplicationExtension               {New-SPWebApplicationExtension}
2 SPWebApplicationHttpTh...               {Disable-SPWebApplicationHttpThrottling, Ena...
2 SPWebApplicationHttpTh...               {Get-SPWebApplicationHttpThrottlingMonitor, ...
3 SPWebPartPack                           {Get-SPWebPartPack, Install-SPWebPartPack, U...
4 SPWebTemplate                           {Get-SPWebTemplate, Install-SPWebTemplate, S...
2 SPWorkflowConfig                        {Get-SPWorkflowConfig, Set-SPWorkflowConfig}
```

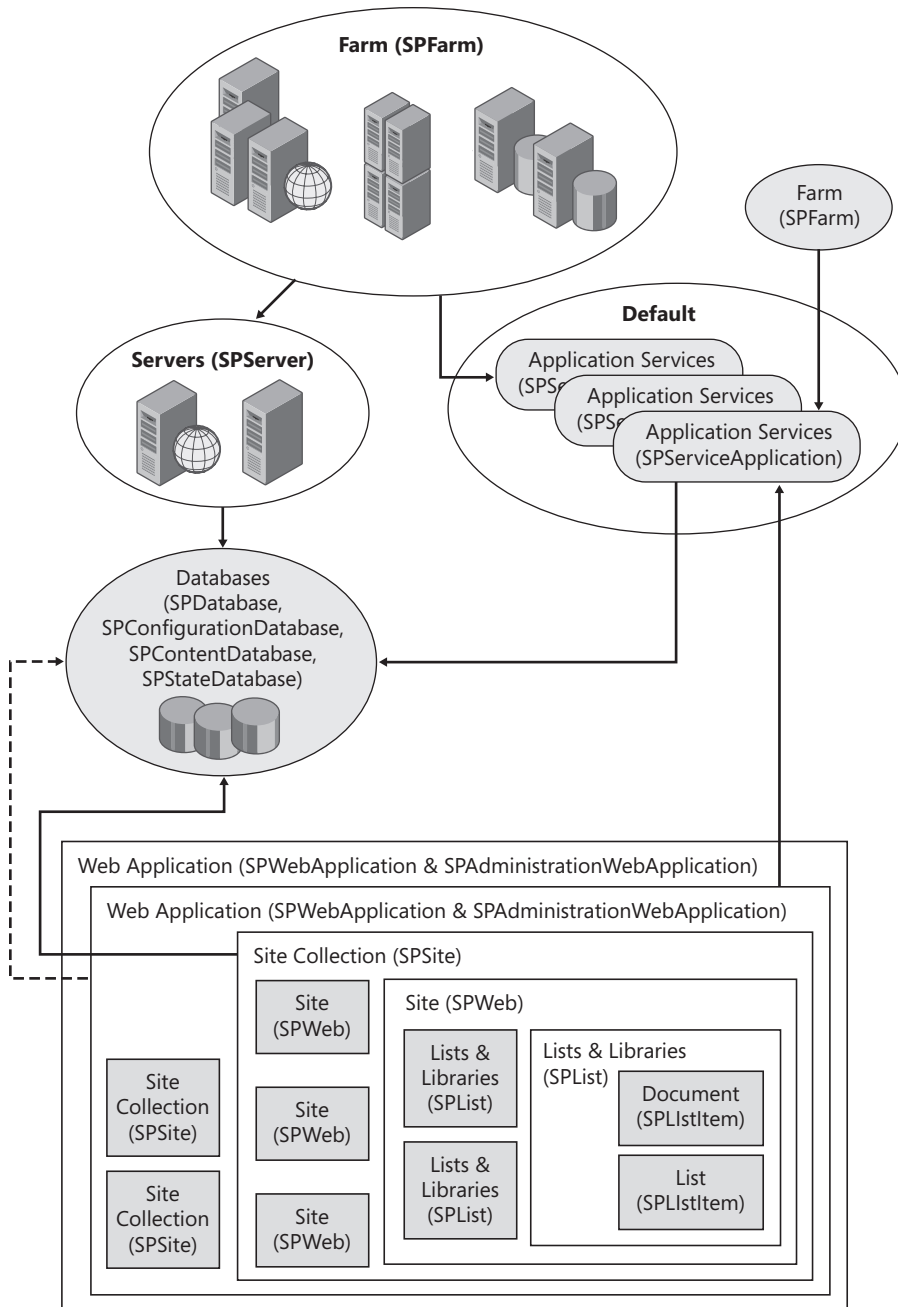


FIGURE 5-3 Common SharePoint objects and their relationships

Understanding Properties and Methods

Everything that SharePoint knows about an object is stored in data elements and values known as *properties*, which also describe the state of that object. For example, the SPContentDatabase object contains all the information about the content databases within the farm.

Objects not only have properties, they also contain actions that determine how they can be manipulated. In object-oriented terminology, these actions, or different ways of acting on an object, are called *methods*. To determine the properties and methods for the SPSite object, use the cmdlets in the following example, where a portion of the output is shown.

```
PS C:\Users\peter> Get-SPSite | Get-Member
```

```
TypeName: Microsoft.SharePoint.SPSite
```

Name	MemberType	Definition
----	-----	-----
AddWorkItem	Method	System.Guid AddWorkItem(Sys...
BypassUseRemoteApis	Method	System.Void BypassUseRemote...
CheckForPermissions	Method	System.Void CheckForPermiss...
Close	Method	System.Void Close()
ConfirmUsage	Method	bool ConfirmUsage()
Delete	Method	System.Void Delete(), Syste...
Dispose	Method	System.Void Dispose()
...		
ToString	Method	string ToString()
UpdateValidationKey	Method	System.Void UpdateValidatio...
VisualUpgradeWebs	Method	System.Void VisualUpgradeWe...
AdministrationSiteType	Property	Microsoft.SharePoint.SPAdmi...
AllowDesigner	Property	System.Boolean AllowDesigne...
AllowMasterPageEditing	Property	System.Boolean AllowMasterP...
...(not all output shown)		
WarningNotificationSent	Property	System.Boolean WarningNotif...
WebApplication	Property	Microsoft.SharePoint.Admini...
WorkflowManager	Property	Microsoft.SharePoint.WorkfL...
WriteLocked	Property	System.Boolean WriteLocked ...
Zone	Property	Microsoft.SharePoint.Admini...

BEST PRACTICE Never miss an opportunity to pipe the Get-SP<noun> cmdlet to the Get-Member cmdlet. It is sometimes surprising what information an object is able to retrieve.



REAL WORLD Performance Issues When Running Windows PowerShell Commands

If you want to find all the site collections in the farm, you can type **Get-SPSite**. However, the only output you will see is the URL of the site collections. For performance reasons, the `Url` property is the only property displayed by default. This is because site collections have many properties; if you display all of these properties for a large number of site collections, you will consume a large amount of memory, and the command will take a long time to run. Displaying just the `Url` property is almost instantaneous because it is cached. Therefore, you should refrain from running this command to display more than a few additional properties.

To display more than the default property, you can use the select Windows PowerShell cmdlet with the pipe (`|`) character. You also can use the wildcard character (`*`) with the select statement to list all the property values, as shown in the following example.

```
PS C:\Users\Peter> Get-SPSite "http://teams" | select *
```

```
ApplicationRightsMask      : FullMask
ID                          : cbf3290e-000e-4768-953c-99a983430283
SystemAccount               : SHAREPOINT\system
Owner                       : CONTOSO\spadmin
SecondaryContact            : CONTOSO\peter
GlobalPermMask              : FullMask
IISAllowsAnonymous         : False
Protocol                    : http:
HostHeaderIsSiteName       : False
HostName                    : teams
Port                        : 80
...(not all output shown)
AllowDesigner               : True
AllowRevertFromTemplate     : False
AllowMasterPageEditing     : False
ShowURLStructure           : False
```

To get a list of all site collections in your farm, together with the name of the content database where they are stored (output that gives only the URL of the site collection and the content database name), you would use the commands as shown in the following example, which also shows the sample output.

```
PS C:\Users\peter> Get-SPSite | select url, contentdatabase
```

Url	ContentDatabase
---	-----
http://MySite	SPContentDatabase Name=Contoso_MySiteDB
http://MySite/personal/peter	SPContentDatabase Name=Contoso_PersonalDB
http://MySite/personal/erin	SPContentDatabase Name=Contoso_PersonalDB
http://intranet.contoso.msft	SPContentDatabase Name=Contoso_IntranetDB
http://teams	SPContentDatabase Name=Contoso_TeamsDB
http://teams/sites/Finance	SPContentDatabase Name=Contoso_TeamsDB

Using Verbs

SharePoint uses common verbs that you have seen with other sets of cmdlets; there are also a number of new verbs added in SharePoint that you may not have seen. Again, you can find these by using the Get-Command cmdlet, as shown here.

```
PS C:\Users\Peter>Get-Command -PSnapin "Microsoft.SharePoint.PowerShell" |  
>>sort verb | group verb | sort count -descending  
>><ENTER>
```

Count	Name	Group
139	Get	{Get-PluggableSecurityTrimmer, Get-SPAccessS...
104	Set	{Set-SPAccessServiceApplication, Set-SPAAlter...
88	New	{New-SPAccessServiceApplication, New-SPAAlter...
77	Remove	{Remove-PluggableSecurityTrimmer, Remove-SPA...
11	Update	{Update-SPFarmEncryptionKey, Update-SPInfoPa...
10	Add	{Add-PluggableSecurityTrimmer, Add-SPClaimTy...
10	Install	{Install-SPApplicationContent, Install-SPDat...
8	Start	{Start-SPAdminJob, Start-SPAssignment, Start...
8	Import	{Import-SPBusinessDataCatalogModel, Import-S...
8	Export	{Export-SPBusinessDataCatalogModel, Export-S...
8	Uninstall	{Uninstall-SPDataConnectionFile, Uninstall-S...
7	Disable	{Disable-SPBusinessDataCatalogEntity, Disabl...
6	Enable	{Enable-SPBusinessDataCatalogEntity, Enable-...
6	Clear	{Clear-SPLogLevel, Clear-SPMetadataWebServic...
5	Stop	{Stop-SPAssignment, Stop-SPEnterpriseSearchQ...
4	Move	{Move-SPBlobStorageLocation, Move-SPProfileM...
3	Initialize	{Initialize-SPContentDatabase, Initialize-SP...
3	Upgrade	{Upgrade-SPContentDatabase, Upgrade-SPEnterp...
3	Backup	{Backup-SPConfigurationDatabase, Backup-SPFa...
3	Restore	{Restore-SPEnterpriseSearchServiceApplicatio...
2	Resume	{Resume-SPEnterpriseSearchServiceApplication...
2	Test	{Test-SPContentDatabase, Test-SPInfoPathForm...
2	Suspend	{Suspend-SPEnterpriseSearchServiceApplicatio...

2	Revoke	{Revoke-SPBusinessDataCatalogMetadataObject, ...}
2	Mount	{Mount-SPContentDatabase, Mount-SPStateServi...
2	Grant	{Grant-SPBusinessDataCatalogMetadataObject, ...}
2	Dismount	{Dismount-SPContentDatabase, Dismount-SPStat...
1	Merge	{Merge-SPLogFile}
1	Receive	{Receive-SPSharedServiceApplicationInfo}
1	Disconnect	{Disconnect-SPConfigurationDatabase}
1	Unpublish	{Unpublish-SPServiceApplication}
1	Connect	{Connect-SPConfigurationDatabase}
1	Rename	{Rename-SPServer}
1	Restart	{Restart-SPEnterpriseSearchQueryComponent}
1	Copy	{Copy-SPBusinessDataCatalogACLToChildren}
1	Ping	{Ping-SPEnterpriseSearchContentService}
1	Publish	{Publish-SPServiceApplication}

As you can see, the most common SharePoint-related verbs are

- **Get** Queries a specific object or a collection of object and retrieves information concerning that object. For example, you could use it to retrieve information on all service applications. Only Get commands return one or more objects; all other commands execute on one object at a time.

```
Get-SPServiceApplication
```

- **Set** Modifies specific settings of an object. For example, you could use it to set the site collection owner for a specific site collection.

```
Set-SPSite http://teams -OwnerAlias contoso\peter
```

- **New** Creates a new instance of an object, such as creating a new site collection.

```
New-SPSite http://teams/sites/HR -OwnerAlias contoso\peter
-Name "HR Team" -Template STS#0
```

- **Remove** Removes (deletes) an instance of the object, such as a site collection and all its subsites. Because of the dangerous nature of this verb, you usually have to confirm this action by using the *-confirm* parameter. If this parameter is not present, you will see a message asking you to confirm the action, as shown here.

```
PS C:\Users\Peter> Remove-SPSite http://teams/sites/HR
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-SPSite" on Target "http://teams/sites/Finance".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
(default is "Y"):y
```

Parameters are used throughout the examples shown here. These are the words prefixed with a hyphen (-). In the next section, you will look at the Get-Help cmdlet and learn how it can help you find out more about these parameters, including which parameters are available for use with different cmdlets.

Getting Help

Windows PowerShell contains an extensive built-in help system, and you can access it quickly by typing **help** at the command-line interface. This is an alias for the cmdlet `Get-Help`. You can even get help about `Get-Help` by typing **Get-Help Get-Help**. The output from such a command lists a description of the cmdlet, the syntax used by the cmdlet, parameter descriptions, examples, and other related notes. You also can use the `Get-Help` cmdlet to get help with the basic Windows PowerShell language. For example, you can use the following commands to get help about specific topics: `Get-Help foreach`, or `Get-Help substring`, or `Get-Help variables`.

NOTE The best way to learn Windows PowerShell is to use its built-in help system. If you or your developers create new Windows PowerShell cmdlets, make sure you create your own help files. In addition to the built-in help, there are many resources on the Internet. For example, you can download a free Windows PowerShell Help 2.0 for either 32-bit or 64-bit from <http://www.primaltools.com/downloads/communitytools/>.

Help Files

`Get-Help` finds the information to display by using Extensible Markup Language (XML) files that are installed on the server. When you installed SharePoint, help files were installed along with the Windows PowerShell snap-in. These files can be found in `%CommonProgramFiles%\Microsoft Shared\Web Server Extensions\14\CONFIG\PowerShell\Help` or `%CommonProgramFiles%\Microsoft Shared\Web Server Extensions\14\CONFIG\PowerShell\Help\locale`, where *locale* defines the language of the computer, such as `en-us`. These files use the naming convention `<dll filename>-help.xml`, such as `Microsoft.SharePoint.PowerShell.dll-help.xml`, `Microsoft.SharePoint.Search.dll-help.xml`, and `Microsoft.Office.Access.Server.dll-help.xml`. You can find which cmdlets use which XML help file by typing

```
PS C:\Users\Peter>Get-Command -PSSnapin "Microsoft.SharePoint.PowerShell" |
>>sort helpfile, name |
>>Format-Wide name -column 2 -groupby helpfile | more
>><ENTER>
```

```

:
Get-SPEXcelBlockedFileType           Get-SPEXcelDataProvider
Remove-SPEXcelDataProvider           Remove-SPEXcelUserDefinedFunction
Set-SPEXcelFileLocation

HelpFile: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions
\14\CONFIG\PowerShell\Help\microsoft.office.access.server.dll-help.xml
```

```
Get-SPAccessServiceApplication      New-SPAccessServiceApplication
Set-SPAccessServiceApplication
```

```
HelpFile: C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions
\14\CONFIG\PowerShell\Help\Microsoft.Office.Excel.Server.MossHost.dll-help.xml
```

```
Get-SPExcelDataConnectionLibrary  Get-SPExcelFileLocation
Get-SPExcelServiceApplication      Get-SPExcelUserDefinedFunction
New-SPExcelBlockedFileType         New-SPExcelDataConnectionLibrary
New-SPExcelDataProvider            New-SPExcelFileLocation
New-SPExcelServiceApplication      New-SPExcelUserDefinedFunction
Remove-SPExcelBlockedFileType      Remove-SPExcelDataConnectionL...
Remove-SPExcelFileLocation         Set-SPExcelDataConnectionLibrary
Set-SPExcelDataProvider            Set-SPExcelServiceApplication
-- More --
```

MORE INFO You can find more information about the format of the cmdlet XML help files and how to make your own help files to complement new cmdlets you create at <http://blogs.msdn.com/powershell/archive/2006/09/14/Draft-Creating-Cmdlet-Help.aspx> and <http://blogs.msdn.com/powershell/archive/2008/12/24/powershell-v2-external-maml-help.aspx>.

ISE Help

In addition to the command-line help function, there is a graphical help file (WindowsPowerShellHelp.chm) located at %SystemRoot%\Help\mui\<LCID> where *LCID* is the locale identifier or language of your installation, such as 0409. You can access the graphical help file when using the ISE by selecting the command you want help for and then pressing F1. The chm file opens, and help is displayed for the command, as shown in Figure 5-4.

However, the chm file contains help only for the core Windows PowerShell cmdlets. If you need SharePoint-specific help, you should use the SharePoint 2010 Management Shell and the Get-Help cmdlet.

MORE INFO For more information on the Help feature in Windows PowerShell and how you can configure it to show help for aliases or how to disable local help and force ISE to get help directly from TechNet, see <http://blogs.microsoft.co.il/blogs/scriptfanatic/archive/2009/01/31/using-help-in-powershell-ise.aspx>. Microsoft recommends that you always check the online help to find the latest information and examples. The link to the online version can be found in the Related Links section when you use the Get-Help command.

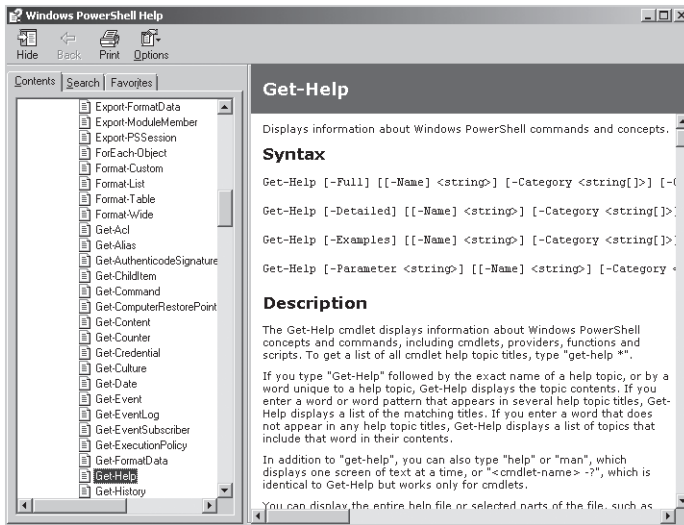


FIGURE 5-4 Windows PowerShell graphical help

Using Parameters

To find which parameters you can use with a cmdlet, type either **Get-Help <cmdlet>** or **Get-Command <cmdlet> -syntax**, where *<cmdlet>* is the name of the cmdlet you want parameter information about. For example, `Get-Help New-SPSite` or `Get-Command New-SPSite -syntax` will provide you with information about the parameters associated with `New-SPSite`. The `Get-Help` cmdlet also has several parameters associated with it, and depending on the amount of information you want, you can use them with the `Get-Help` command. If you know a cmdlet very well and do not want a descriptive explanation of what the cmdlet does, then you may simply want to see some examples of how to use the cmdlet, in which case you would type **Get-Help New-SPSite -examples**.

To learn about which parameters to use with a cmdlet, review the syntax section of the help results, as shown in the following output. Optional parameters are enclosed within square brackets ([]) and the values to be passed to the parameters are enclosed within angled brackets (<>). The following example shows the syntax section of the `Get-SPSite` cmdlet.

SYNTAX

```
Get-SPSite [-AssignmentCollection <SPAssignmentCollection>]
[-Confirm [<SwitchParameter>]] [-Filter <ScriptBlock>] [-Limit <String>]
[-WebApplication <SPWebApplicationPipeBind>] [-WhatIf [<SwitchParameter>]]
[<CommonParameters>]
```

```

Get-SPSite -Identity <SPSitePipeBind> [-AssignmentCollection
<SPAssignmentCollection>] [-Confirm [<SwitchParameter>]] [-Filter <ScriptBlock>]
[-Limit <String>] [-Regex <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
[<CommonParameters>]

Get-SPSite -ContentDatabase <SPContentDatabasePipeBind> [-AssignmentCollection
<SPAssignmentCollection>] [-Confirm [<SwitchParameter>]] [-Filter <ScriptBlock>]
[-Limit <String>] [-WhatIf [<SwitchParameter>]] [<CommonParameters>]

Get-SPSite -SiteSubscription <SPSiteSubscriptionPipeBind> [-AssignmentCollection
<SPAssignmentCollection>] [-Confirm [<SwitchParameter>]] [-Filter <ScriptBlock>]
[-Limit <String>] [-WhatIf [<SwitchParameter>]] [<CommonParameters>]

```

Notice that there are four ways to use Get-SPSite, and each syntax has an *-AssignmentCollection* parameter. The *-AssignmentCollection* parameter relates to an important aspect of the built-in SharePoint cmdlet, which is examined in more detail in the sidebar titled “Memory Considerations When Using Windows PowerShell,” which appears later in this chapter.

The first syntax example shown in the listing above has no mandatory parameter, so the Get-SPSite cmdlet would have a scope of the farm or Web application. The other three syntaxes have a mandatory parameter that defines the scope as site collection, content database, or collection of site collections that subscribe to shared settings, features, and services, as shown in the following code. As is the case with the other SPSite cmdlets, in a large installation, this cmdlet could output a large amount of information, and the queries to many content databases could severely affect the performance of the SharePoint servers. Therefore, by default, the output reports only on the first 20 objects.

You can alter this output limit by using the *-limit* parameter, as shown in the following example, which outputs all site collections that are stored in the Contoso_TeamsDB content database.

```
Get-SPSite -ContentDatabase Contoso_TeamsDB -limit all
```

You can provide the values for these scope parameters, or in the case of the *-identity* parameter, where the value is a URL, you can use the wildcard character (*). The default help with no parameter provides a limited description of the command.

For more information, use the *-detailed* or the *-full* parameter. The *-full* parameter provides additional information about the parameters. For example, additional information about the *-identity* parameter is shown on the following page.

PARAMETERS

`-Identity <SPSitePipeBind>`

Specifies the URL or GUID of the site collection to get.

The type must be a valid URL, in the form `http://server_name` or `http://server_name/sites/sitename`, or a valid GUID (for example, `12345678-90ab-cdef-1234-567890bcdefgh`).

Required?	true
Position?	1
Default value	
Accept pipeline input?	True
Accept wildcard characters?	false

If you look at the last five lines of the output, you see that it provides more information on how to use the `-identity` parameter. For example, the required value is equal to `True`, which indicates that the `-identity` parameter is a mandatory parameter for the `Get-SPSite` cmdlet. However, the position value is equal to `1`, which means that if you place a URL of a site collection immediate after the `Get-SPSite` cmdlet, you do not have to use the `-identity` parameter because the URL will be read as the identity (URL) of a site collection, where all the property values for site collections that exist under the personal managed path are displayed, as shown in the following example.

```
Get-SPSite http://MySite/personal/* | Select *
```

Many of the parameters have a value enclosed in angle brackets, with the suffix `PipeBind`. This indicates that the parameter can accept an object variable of a specific type. For example, `[-ContentDatabase <SPContentDatabasePipeBind>]` means that the `-ContentDatabase` parameter can accept a SharePoint content database object, and this object can be passed to the `ContentDatabase` object either as a variable, or it can be piped in from the results of another cmdlet. When you pipe an object into a cmdlet, you do not need to type or use a variable. When you pipe an object from one cmdlet into another cmdlet, you do not have to specify the parameter, because Windows PowerShell checks the type of object and will match it to the correct parameter. The following example lists all site collections that are stored in any content databases that have the characters `oso` anywhere in their names.

```
Get-SPContentDatabase | where {$_.name -match "oso"} | Get-SPSite -Limit 50
```

`Get-SPContentDatabase` returns a collection of content databases, and then you need to check each content database to see if it matches our condition. The `Where-object` cmdlet, which has the aliases `where` or `?`, acts like a loop, so it will check each content database object returned from the `Get-SPContentDatabase` cmdlet. The variable `$_` represents one content database object, and `.name` is the content database property that needs to be matched for our query. The operator is `-match`.

Windows PowerShell has a number of operators, such as *-eq* (equal to), *-ne* (not equal to), *-lt* (less than), *-ge* (greater than or equal to), *-like* (matches a wildcard pattern), and *-notlike* (does not match a wildcard pattern). The *-like*, *-notlike*, *-match*, and *-notmatch* operators are used for pattern matching. The *-match* and *-notmatch* operators use regular expressions to determine whether a value does or does not contain a match for an expression.

MORE INFO For more information on Windows PowerShell operators, see the section titled “Working with Expressions and Operators” in Chapter 5, “Navigating Core Windows PowerShell Structures,” in *Windows PowerShell 2.0, Administrator’s Pocket Consultant*, by William Stanek (Microsoft Press, 2009).

An alternative method to use instead of a WHERE statement is the *-filter* parameter, which uses the same syntax as the WHERE statement and can produce the same results. However, the *-filter* parameter executes on the server. Piping the results of the `Get-SPContentDatabase` cmdlet to the WHERE statement causes a SQL round trip to wherever the Windows PowerShell client is running. Using the *-filter* parameter can result in a faster performance, allowing the command to leverage the filtering abilities of SQL instead of attempting a local search. However, when you use the *-filter* parameter, only the Owner, SecondaryOwner, and Lock-State properties can be accessed for SPSite cmdlets. A number of cmdlets have other parameters that allow you to filter the resulting output. For example, `SPWeb` provides parameters to filter output on the template or title of the site (web). Still, in many cases the *-filter* parameter can save you time—for example, finding all the blog sites in a farm of 4300 webs takes about 1.2 seconds with the *-filter* parameter, but it takes approximately 15 minutes if you use a WHERE statement. Following are several examples that use the *-filter* parameter.

- This example returns all site collections whose primary owner has the username peter.

```
Get-SPSite -Filter {$_.Owner -eq "contoso\peter"}
```

- This example returns all websites in the `http://teams/sites/HR` site collection that were created using the Blank Meeting Workspace template.

```
Get-SPSite http://teams/sites/HR | Get-SPWeb -Filter {$_.Template -eq "STS#03"}
```

NOTE When you create a new site collection, if you do not specify a template or the template cannot be found, then you will need to create a website as the root of the site collection. You can do this by using the `New-SPWeb` command or by selecting the appropriate site template when you display the site collection for the first time in the browser.

To find details of the templates installed, together with the appropriate template ids, use the `Get-SPWebTemplate` cmdlet. A sample of the output of this command is shown here.

Name	Title	LocaleId	Custom
GLOBAL#0	Global template	1033	False
STS#0	Team Site	1033	False
STS#1	Blank Site	1033	False
STS#2	Document Workspace	1033	False
MPS#0	Basic Meeting Workspace	1033	False
MPS#1	Blank Meeting Workspace	1033	False
MPS#2	Decision Meeting Workspace	1033	False
MPS#3	Social Meeting Workspace	1033	False
MPS#4	Multipage Meeting Workspace	1033	False
CENTRALADMIN#0	Central Admin Site	1033	False
WIKI#0	Wiki Site	1033	False
BLOG#0	Blog	1033	False
SGS#0	Group Work Site	1033	False
TENANTADMIN#0	Tenant Admin Site	1033	False
ACCSR#0	Access Services Site	1033	False
ACCSR#1	Assets Web Database	1033	False
ACCSR#3	Charitable Contributions Web Database	1033	False

On the `Get-SPSite` and `Get-SPSiteAdministration`, provide two other filters that use the SQL-driven filtering: the “Wildcard URLs” and regular expression (`-Regex`) parameter, for example:

```
Get-SPSite http://intranet/sites/*
Get-SPSite "http://intranet/(sites|teams)/HR" -Regex
```

Memory Considerations When Using Windows PowerShell

The default behavior of Windows PowerShell is that it runs a multithread environment and that each line, function, or script runs in its own thread (`$host.Runspace.ThreadOptions == "Default"`). This can cause memory leaks, but as you saw in the section titled “Working with the SharePoint 2010 Management Shell” earlier in this chapter, the SharePoint 2010 Management Shell runs each line, function, or script in the first thread (`$host.Runspace.ThreadOptions = "ReuseThread"`), which mitigates the problem—but memory leaks can still occur. Hence, the two SharePoint cmdlets that you should learn to use are the `Start-SPAssignment` and the `Stop-SPAssignment`. These relate to the `-SPAssignmentCollection` parameter that you may see used with a number of SharePoint cmdlets to return a “disposable object.” In developer-speak, these are objects that implement the `IDisposable` Interface. If these objects are not disposed of correctly, they can cause memory

leaks. In particular, this relates to the `SPSite`, `SPWeb`, and `SPSiteAdministration` objects. If you use cmdlets associated with these objects in one pipe and don't use variables to store the objects, the memory used is automatically disposed of at the end of a pipe. If you have a long pipe that obtains many objects, this can cause a shortage of memory on your SharePoint server, which then can have a noticeable effect on requests for pages from that server.

When you store any of the three objects in a variable, you then have to dispose of the memory assigned to that object. The following examples show the main methods of using these objects with variables and how to dispose of the memory.

- Use the simple method of using the `SPAssignment` cmdlets, in which all objects are kept in a global memory store that is released when the `Stop-SPAssignment` cmdlet is called.

```
Start-SPAssignment -Global
$sc = Get-SPSite http://intranet
$sc.Title
Stop-SPAssignment -Global
```

- Use the advanced method of using the `SPAssignment` cmdlet and tell it to track memory assigned to a specific variable. In the following example, the `Start-SPAssignment` cmdlet creates a named store that is pointed to by `$o`, and the variable `$sc` is associated with that named store and populated with information about the Internet site collection object. The `Stop-SPAssignment` cmdlet then releases the memory associated with the named store.

```
$o = Start-SPAssignment
$sc = $o | Get-SPSite http://intranet
Get-SPSite -Limit all
$sc.Title
$o | Stop-SPAssignment
```

- If you are used to developing on the SharePoint platform, then you can use a similar technique to that which you use in your code to overcome this issue. For example, use variables within one Windows PowerShell line and dispose of them at the end of that line.

```
$sc = New-SPSite("http://Intranet"); $sc.Title; $sc.Dispose()
```

This stores the site collection object in the variable `$sc`. The next command then prints out the title of the intranet site collection, and the last command releases the memory that stored the site collection object.

NOTE This same command can be run safely using the following syntax.

```
New-SPSite(http://intranet) | Select Title
```

Be aware that the simple method of calling SPAssignment shown here also can cause memory problems. Any object you obtain between the Stop and Stop SPAssignment commands will be retained in the global assignment store and will not be released until the Stop-SPAssignment cmdlet is executed, whereas when you use the advanced method of calling SPAssignment, only the memory associated with variables you ask SPAssignment to track will be retained until the Stop-SPAssignment cmdlet is executed. Therefore, if you need to view large amount of information as you complete a task, and you do not want to starve the SharePoint server of memory, the advanced method of using SPAssignment is preferable. Use Get-Help About_SPAssignment to find more information.

cmdlets for SharePoint Foundation 2010

SharePoint Foundation 2010 contains over 240 SharePoint-related cmdlets. The exact number can be found by typing the following command.

```
@(Get-Command -PSSnapin "Microsoft.SharePoint.PowerShell").count
```

The most common objects that these cmdlets manipulate are the SPSite, SPSEver, SPWeb, SPBusinessDataCatalogue, and SPConfigurationDatabase objects. Because Windows PowerShell is mainly an administrator's tool, and these are the components an administrator manages, this spread of cmdlet is not unexpected. These cmdlets also can be found in SharePoint 2010, which is built on top of SharePoint Foundation.



ON THE COMPANION MEDIA The document "Microsoft SharePoint Foundation 2010 Cmdlet Reference" is included on this book's companion media. This document contains the output that you would see if you typed the following command in the SharePoint 2010 Management Shell.

```
PS C:\Users\Peter>Get-Command -PSSnapin Microsoft.SharePoint.PowerShell |  
>>Sort Noun, Verb | Get-Help -detailed > CmdletHelp.txt  
>><ENTER>
```

This command generates a list of cmdlet Help topics that is sorted by noun. This list contains the same Help for each cmdlet that you can view in the SharePoint 2010 Management Shell by typing the following command.

```
Get-Help <Cmdlet-Name> -detailed
```

The document contains help information that was current at the time that the document was produced. Microsoft recommends that you always check the online help to find the latest information and examples. You can find the link to the online version in the Related Links section when you use the Get-Help command.

cmdlets for SharePoint 2010

If you use the same Windows PowerShell command shown in the previous section to determine the number of cmdlets available in SharePoint 2010, you will find that there are more than 530 SharePoint-related cmdlets associated with SharePoint 2010. Specifically, that is the number of cmdlets associated with a full install of the Enterprise Edition of SharePoint 2010 without FAST Search installed. To identify the additional cmdlets that SharePoint provides, Windows PowerShell can help again. On a computer with both SharePoint Foundation and a version of SharePoint Server installed, type the following command, changing the name of the redirect file name to reflect the SharePoint installation.

```
PS C:\users\peter>Get-Command -PSSnapin "Microsoft.SharePoint.PowerShell" |
>> Sort noun, verb |group -Property noun -NoElement > cmdlet_sps.txt
>><Enter>
```

Copy the files to the same server and then type the following command.

```
PS C:\Users\Peter>Compare-Object -ReferenceObject $(Get-Content .\cmdlet_sps.txt) '
>>-DifferenceObject $(Get-Content .\cmdlet_spf.txt)
>><ENTER>
```

You will see output similar to the following example.

InputObject	SideIndicator
8 SPContentDatabase	=>
3 PluggableSecurityTrimmer	<=
3 SPAccessServiceApplica...	<=
9 SPContentDatabase	<=
5 SPContentDeploymentJob	<=
4 SPContentDeploymentPath	<=
4 SPDataConnectionFile	<=
1 SPDataConnectionFileDe...	<=
2 SPEnterpriseSearchAdmi...	<=
3 SPEnterpriseSearchLang...	<=
4 SPEnterpriseSearchMeta...	<=
...	

Only those objects that do *not* appear on both installations, and those for which the count of the number of cmdlets per object is different, will appear in this list. The SideIndicator column indicates from which file the object with its cmdlet count came.

In SharePoint 2010, the most common objects that are manipulated by cmdlets are similar to those in SharePoint Foundation. However, the main additional commands available in SharePoint 2010 are the 131 cmdlets that help you manage the search process. (There are only four cmdlets associated with searches for SharePoint Foundation). Type the following command to check this for yourself.

```
PS C:\Users\Peter>@(gcm -PSSnapin "Microsoft.SharePoint.PowerShell" |
>>where {$_.name -like "*search*"}).count
>><ENTER>
```

SharePoint 2010 contains more application services than SharePoint Foundation, such as Excel Services, InfoPath Services, Secure Storage, State Service, Microsoft Visio, and PerformancePoint. Therefore, on a SharePoint Server farm, there are more cmdlets that you can use to create, manipulate, and delete application service objects.

Performing Basic Administrative Tasks

There are two types of administrative tasks that are needed in a SharePoint installation: tasks that need to be completed once and from any server, such as creating a new Web application, configuring the primary owner of a site collection, or deleting a specific website; and tasks that need to be completed more than once on a specific server or on more than one server, such as starting a service on a server, adding or removing a server from the farm, or connecting the server to a specific configuration database. The majority of work you will need to complete as a SharePoint administrator is of the first type—it does not matter which server you are logged on to, because you can complete your work from any server.

Windows PowerShell makes it easier to complete both types of administrative tasks. You can loop around a large number of objects and incorporate the scripts into automated tasks so that administrative tasks can be completed in less time. Just because administrative tasks may be easy to complete using Windows PowerShell, however, you should not allow its use to circumvent the controls you may have in place. Some of these farm-wide tasks are very dangerous; they can affect the whole installation. In a production environment, such tasks should be under strict change management.

BEST PRACTICE Many organizations allow administrative tasks to be completed only by using remote access to the production servers. Computer room access is strictly controlled, so interactive access at the server console is rarely allowed—usually only for hardware-related issues. In such environments, even remote access to the production servers is restricted to a small number of Administrative computers, which administrators have to connect to using a VPN or Remote Desktop. An Administrative computer should be configured to allow only certain users or IP addresses to instigate remote desktop sessions. If this is your scenario, remotely managing your SharePoint Installation using Windows PowerShell should also be restricted to that Administrative computer. Windows PowerShell has a number of built-in capabilities that can help you reduce the risk of tasks that affect the whole of a SharePoint installation; these were referred to in the section titled “Managing Systems Remotely with WinRM” earlier in this chapter.

Windows PowerShell provides a number of “voluntary” capabilities that could be classified as best practices. In the next sections, you will learn more about two of these, the *-whatif* parameter and transcripts.

Using the *-whatif* Parameter

Windows PowerShell is a powerful tool, and like any other scripting language, it is all too easy to borrow someone else's code or download snippets from the Internet. You might then execute them without knowing exactly what they do. Using signed scripts and the Windows PowerShell execution policy, as well as restricting who is allowed to load script files onto your SharePoint server, are ways that you can protect your resources.

Windows PowerShell provides a mechanism that allows you to try a command before you execute it—the *-whatif* parameter. Type the following command.

```
PS C:\Users\Peter> Get-SPSite http://teams/sites/* | Remove-SPSite -whatif
```

This command produces the following sample output that lists all site collections and the operation(s) that would be performed on these objects if you executed the command without the *-whatif* parameter.

```
What if: Performing operation "Remove-SPSite" on Target "http://teams/sites/Sales".
What if: Performing operation "Remove-SPSite" on Target "http://teams/sites/Blogs".
What if: Performing operation "Remove-SPSite" on Target "http://teams/sites/Finance".
What if: Performing operation "Remove-SPSite" on Target "http://teams/sites/Wikis".
What if: Performing operation "Remove-SPSite" on Target "http://teams/sites/HR".
```

Nothing has been deleted by trying the command with the *-whatif* parameter. The parameter tells the `Remove-SPSite` cmdlet to display the object that would be affected by the command without performing the command. In this example, it displays the objects that would be permanently deleted.

Generating Transcripts

A pair of Windows PowerShell cmdlets, `Start-Transcripts` and `Stop-Transcripts`, allows you to record all the commands that you type at the prompt, together with the output. All activity is recorded between the start and stop commands. If you do not type **stop**, the transcript is terminated when the console session is terminated. This could be very useful in your production environment, where the machine-wide profile contains the `Start-Transcript` cmdlet to record the Windows PowerShell activities of all users. The recording to the transcript file is complete when they exit the console.

IMPORTANT Don't use a Windows PowerShell cmdlet or script for the first time in a production environment. Also, try to keep it simple—don't use or create aliases if scripts are going to be supported by administrators who are new to Windows PowerShell, and always use Windows PowerShell comments when you create or amend scripts.



REAL WORLD What If You Don't Use the *-whatif* Parameter?

Sometimes even experienced IT professionals can use the built-in cmdlets incorrectly with serious consequences, especially when cmdlets are used in combination with the pipe character. For example, if you type `Get-SPSite | Remove-SPSite`, all site collections in your farm will be permanently deleted. Everything—all the data in your lists and libraries, and all the content in your websites—would simply be gone. One likely result of this action is that all your help desk telephone lines would immediately ring. If one of the websites that disappeared was your main e-commerce Internet website, your company could quickly lose a great deal of money, and perhaps more importantly, the company could lose potential customers who might never return. To restore your farm, you would need your latest good backup tapes. The lesson of this example is simple: always use the *-whatif* parameter first!

Examples of SharePoint Administrative Tasks

Many administrative tasks must be completed on a daily basis, some are needed only occasionally, and then there are those few that are rarely required. Windows PowerShell can help with all of these types of tasks; in fact, you may discover that it is more critical to script a rarely executed task than one you perform every day. This may be because you need to execute that rare task quickly and correctly, because the ramifications of making even a small mistake while completing the task could be catastrophic. Throughout this book, you will see many examples of when and how to use Windows PowerShell. In the following sections, you explore a few examples that you may not have seen elsewhere.

SEE ALSO There is a section of the Microsoft Script Center, <http://gallery.technet.microsoft.com/ScriptCenter/en-us>, that contains SharePoint-related scripts.

Deploying SharePoint 2010 with Windows PowerShell Scripts

During the lifetime of your SharePoint installation, there may be many tasks that you must complete (hopefully) only once, such as installing SharePoint and creating your farm. In large organizations with a variety of environments and many developers to support, however, you may find it prudent to develop scripts right from the outset that will build your SharePoint Web front-end (WFE) servers and your SharePoint applications servers, which host your applications services, such as InfoPath Services and search query and indexing services. In previous versions of SharePoint, you could complete such tasks using batch files and the SharePoint

command-line tools `psconfig` and `stsadm`. You can still use these tools to automate the installation of your SharePoint 2010 and SharePoint Foundation 2010 servers. You will find more information about using these tools in Chapter 4, "Installing SharePoint 2010," so the details aren't repeated here. Microsoft has developed a script that is really the best practice for creating a SharePoint farm from scratch using Windows PowerShell. The order of these commands and parameters can be quite tricky, so Microsoft recommends that you use the Windows PowerShell module, `SPModule`, documented in the Microsoft's TechNet library article, "Install SharePoint Server 2010 by using Windows PowerShell", at <http://technet.microsoft.com/en-us/library/cc262839.aspx>.

MORE INFO For information about SharePoint automation and the use of `psconfig`, `stsadm`, and Windows PowerShell, see <http://stsadm.blogspot.com/>.

When you work on a SharePoint installation, you will need to merge the logs. The Windows operating systems provide you with event logs to help you with your administrative tasks, and you can use the Windows PowerShell `Get-Eventlog` cmdlet to obtain detailed information from those logs. SharePoint also provides the Unified Logging Service (ULS). The ULS contains all application log events, and third-party logging software can be integrated into them as well. SharePoint 2010 includes several Windows PowerShell cmdlets for retrieving information and configuring the ULS. To find how ULS is configured on your farm, type the following command. (The sample output for this command is also shown here.)

```
Get-SPDiagnosticConfig
```

```
AllowLegacyTraceProviders           : False
CustomerExperienceImprovementProgramEnabled : True
ErrorReportingEnabled               : True
ErrorReportingAutomaticUploadEnabled  : True
DownloadErrorReportingUpdatesEnabled  : True
DaysToKeepLogs                     : 14
LogMaxDiskSpaceUsageEnabled          : True
LogDiskSpaceUsageGB                 : 3
LogLocation                         : %CommonProgramFiles%\Microsoft Shared\Web Server Extensions\14\LOGS\
LogCutInterval                      : 30
EventLogFloodProtectionEnabled       : True
EventLogFloodProtectionThreshold     : 5
EventLogFloodProtectionTriggerPeriod : 2
EventLogFloodProtectionQuietPeriod   : 2
EventLogFloodProtectionNotifyInterval : 5
ScriptErrorReportingEnabled          : True
ScriptErrorReportingRequireAuth      : True
ScriptErrorReportingDelay            : 60
```

When diagnosing a problem, you can use cmdlets associated with the `SPLogEvent` object. You need machine administrator privileges to run these cmdlets, so you may need to start SharePoint 2010 Management Shell using Run As Administrator. Then you would use commands in steps similar to those shown in the following example.

1. Set up two variables to store the start day and the end date, for example.

```
$SDate = Get-Date -Day 13 -Month 04 -Year 2010
$EDate = Get-Date -Day 14 -Month 04 -Year 2010
```

Or the date could be today's date, but the event occurred approximately 15 minutes earlier than the current time. In this case, you set the variable to 20 minutes before the current time and 10 minutes before the current time.

```
$SDate = (Get-Date).AddMinutes(-20)
$EDate = (Get-Date).AddMinutes(-10)
```

2. Obtain a list of events between those two dates and times.

```
Get-SPLogEvent -StartDate $SDate -EndTime $Edate
```

If the `-EndTime` parameter is not provided, the event logs will be displayed up until the current date and time.

A tracking number, known as the Correlation ID, is associated with each request you make to SharePoint. This number is stored as a value in a property associated with the `SPLogEvent` object. SQL Profiler traces also show the Correlation IDs. This number is displayed on an Error page, as shown in Figure 5-5.

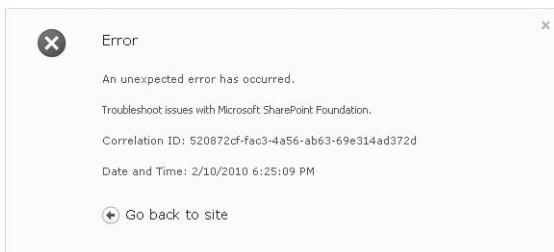


FIGURE 5-5 Error page showing the Correlation ID

In a large or heavily used SharePoint installation, many logs may be produced during the period when a problem occurs. You can use the Correlation ID to help troubleshoot errors by reducing the amount of log information that is returned to those entries that are associated with the incident. Again, set the `$sdate` and the `$edate` variables to reflect the time around the incident. For an incident that you can't reproduce and one of your users always can reproduce, you might tell the help desk to ask the user to take a screenshot of the error page so

you can be sure to obtain the exact Correlation ID number. Then you can pipe the results of the command shown in step 2 to the Where-Object cmdlet, as shown in the following example.

```
PS C:\Users\Peter>Get-SPLogEvent -StartDate $SDate -EndTime $EDate |  
  where {$_.Correlation -eq "68bc5cf4-5a8c-4517-a879-86e35e57c862"}
```

The output will display only event logs between the two dates that are related to the problem associated with that Correlation ID.

Managing SharePoint Services

After you have run the SharePoint 2010 Products Configuration Wizard to install SharePoint on your servers, you can use the SharePoint 2010 Central Administration website to launch the Farm Configuration Wizard. Although it will get you up and running very quickly, it might not configure your farm as you want; for example, the database names it creates will be in the format *name_GUID*, where *GUID* is a randomly generated, globally unique identifier. Also, the farm wizard uses the account entered as the Application Pool Identity for the default content Web application and starts the Web Analytics Service by default—both of which may not be desirable on a production farm.

Using the wizard may be sufficient if you want to create a quick prototype or perhaps if you are using SharePoint in a developer environment. However, if any of those environments are limited in their CPU power or in the amount of memory they have, the farm wizard may not be suitable either. For example, the Web Analytics Service uses a large amount of CPU, so your developers might be interested in stopping that service. There are other services that they might want to start and stop on a regular basis, especially if they are using their computers as their day-to-day desktop connections, where they read e-mails and complete documentations, as well as to run SharePoint for development or prototyping.

You can use tools such as the Central Administration website to manage the configuration; however, you may forget to execute all the tasks necessary to complete a configuration change. Instead, you can use Windows PowerShell scripts, which—after you have tested them to make sure that you have not forgotten any tasks—should make managing configuration changes more reliable and error free. You should store these scripts in a central location after they are developed, to make them convenient for fellow administrators to use and incorporate in their environments. Your developer may also thank you for developing these scripts. Such automated tasks might include the following items.

- To quickly delete a specific Web application, type

```
Get-SPWebApplication http://teams | Remove-SPWebApplication -Confirm
```

- To remove the Web application, the IIS website, and all the associated databases, type

```
Remove-SPWebApplication http://teams -Confirm -DeleteIISite '  
  -RemoveContentDatabases
```

If you do not run the Farm Configuration Wizard, you might find that you need a service that you originally thought you would not need. For example, if no application services are started, and then you try to use one of the SharePoint 2010 workflows, the Microsoft Visio diagram of the workflow progress will not be displayed. However, to solve this problem, you must do more than just create the Visio Application Service. There are other dependencies involved, one of which is the State Service Application. The state services can only be created in the Central Administration site, by using the farm wizard—something you will want to avoid as an administrator. To create the state service using Windows PowerShell, type the following commands.

```
PS: C:\Users\Peter>New-SPStateServiceDatabase -Name Contoso_StateService_DB |
>>New-SPStateServiceApplication -Name Contoso_StateService |
>>New-SPStateServiceApplicationProxy -Name Contoso_StateService_Proxy -Default
>><ENTER>
```

NOTE This command may not work in your environment if the associated service instance is not running. For a more comprehensive script, see “SP+PS 2010: PowerShell to Create a Service Application” at <http://sharepoint.microsoft.com/blogs/zach/Lists/Posts/Post.aspx?ID=50>.

Each service application is slightly different. For example, not all service applications require a database. Chapter 7 “Scaling Out a SharePoint Farm,” contains more information about this. However, each service will require a service application proxy, which connects a Web application to the service application.

Using Windows PowerShell During the Upgrade Process

The following sections provide information about some Windows PowerShell commands you might find useful during an upgrade. The upgrade process is explained in more detail in Chapter 22, “Upgrading to SharePoint 2010.”

Preparing for an Upgrade

After installing SharePoint 2010 and before upgrading, you can use the Test-SPContentDatabase cmdlet to check your SharePoint Server 2007 databases for current or potential issues, such as orphan data, missing site definitions, missing features, or missing assemblies. This cmdlet complements the pre-upgrade checker report mentioned in Chapter 22. Checking for these issues does not disrupt your SharePoint installation, and this cmdlet can also test the status of SharePoint 2010 databases, which can assist you in maintaining the continued health of your environment. The following example shows the command and its sample output.

```
Test-SPContentDatabase -name W_intranet -WebApplication http://www.contoso.msft
```

```
Category      : SiteOrphan
Error         : True
UpgradeBlocking : False
Message       : Database [W_intranet] contains a site (Id = [46ad6d70-9a5c-4de0-8daa-0f73f2466a6a], Url = [/]) whose id is already associated with a different database (Id = [6987d2d8-6291-4ead-9eb0-aefe7097a58e], name = [W_Intranet]) in the site map. Consider deleting one of these sites which have conflicting ids.
Remedy        : The orphaned sites could cause upgrade failures. Try detach and reattach the database which contains the orphaned sites. Restart upgrade if necessary.

Category      : SiteOrphan
Error         : True
UpgradeBlocking : False
Message       : Database [W_intranet] contains a site (Id = [46ad6d70-9a5c-4de0-8daa-0f73f2466a6a], Url = [/]) whose url is already used by a different site, in database (Id = [6063e77c-991f-4c4b-b3ac-68cb62e66502], name = [w_Internet]), in the same web application. Consider deleting one of the sites which have conflicting urls.
Remedy        : The orphaned sites could cause upgrade failures. Try detach and reattach the database which contains the orphaned sites. Restart upgrade if necessary.
```

Performing Post-Upgrade Tasks

The upgrade process is divided into two components, the content database upgrade and the visual upgrade. By separating these two components, your SharePoint team can decide to upgrade the Microsoft Office SharePoint Server 2007 Web application without users noticing the change, because their sites will maintain the old look and feel. Using the browser, they can preview what their site looks like with the new SharePoint 2010 look and feel and then decide when they permanently want to switch to the visual upgrade. After they make the switch, the browser will not allow them to switch back to the SharePoint Server 2007 look and feel.

The Set-SPWeb cmdlet does not have any parameter to help with this task; you will need to use the properties of the SPWeb object to complete this task. Type the following commands if you want to switch sites back to the old look and feel.

```

Start-SPAssignment -Global
$web = Get-SPWeb http://teams ;
$web.UIVersion = 3;
$web.UIVersionConfigurationEnabled = $true;
$web.Update();
Stop-SPAssignment

```

NOTE Previously in this chapter, you created variables to store values and objects. Variable names are prefixed with the dollar sign, \$. In the sample code shown here, you will see *\$true* used. This, together with a number of other keywords that look like variables, is an example of a special variable. A special variable should be treated as a reserved word that you cannot use in your scripts to store values or object. The special variable *\$true* represents the value True, *\$false* represents False, *\$null* represents null, and *\$_* also has been used in this chapter and contains the current pipeline object. It is used in script blocks, filters, and the Where cmdlet.

After you run these commands, the website will revert to the SharePoint Server 2007 look and feel. The Site Actions button will be in the upper-right corner, for example, instead of in the upper-left corner. The Site Actions menu will have the Visual Upgrade option available, although the Visual Upgrade option at the site collection level may not be enabled. To enable the Visual Upgrade option at the site collection level, use the following commands.

```

$site = Get-SPSite http://teams
$site.UIVersionConfigurationEnabled = $true

```

When updating the properties of the SPSite object, there is no update method; the changes to the SPSite properties take effect immediately. You could type the following line as an alternative to the previous two lines of commands.

```

(Get-SPSite http://teams).UIVersionConfigurationEnabled = $false

```

To reset all “team” websites within a site collection back to the SharePoint Server 2007 look and feel, use the following command.

```

PS C:\Users\zzspfarm> Get-SPsite http://teams |
>> Get-SPWeb -Filter {$_.Template -eq "STS#0"} |
>> ForEach-Object {
>> $_.UIVersion =3;
>> $_.UIVersionConfigurationEnabled = $false;
>> $_.Update();
>> }
>><ENTER>

```

NOTE These commands also can be run on a site or site collection that has not been upgraded—that is, on a site that was created as a version 4 site, thereby making a SharePoint 2010 site look like a SharePoint 2007 site.

Summary

This chapter explains how you can become familiar with the new Windows PowerShell cmdlets introduced by SharePoint 2010. It explores the SharePoint 2010 Management Shell and how to accomplish some simple but powerful administrative tasks using one-line Windows PowerShell commands.

There are more than 500 SharePoint cmdlets. This chapter could not possibly explain each command; instead, its aim was to provide some general guidelines and good practices for using SharePoint PowerShell in a production environment, such as the configuration and use of remote management, the voluntary use of transcripts, and the use of the *-whatif* parameter. You should not practice or try out Windows PowerShell commands for the first time on your production environment. Keep your scripts simple, use comments to document your scripts, and don't use or create aliases if the scripts are going to be supported by administrators who are unfamiliar with Windows PowerShell.

The chapter covered two important cmdlets, *Start-SPAssignment* and *Stop-SPAssignment*, in some detail. These cmdlets help with cmdlets that return disposable objects, such as *SPSite*, *SPWeb*, and *SPSiteAdministration*. Other SharePoint cmdlets were introduced to expand your knowledge of Windows PowerShell techniques. Chapter 12 builds on this chapter, using search scenarios. The next chapter continues the discussion of SharePoint 2010 farm management and specifically the Central Administration interface.

Index

Symbols and Numbers

- [] (square brackets), 224
- (hyphen), 221
- \$ (dollar sign), 240
- * (asterisk), 219
- ` (back tick), 582
- | (pipe), 219, 582
- + (plus sign), 1002

A

- AAM. *See* alternate access mappings (AAM)
- access control list (ACL), 895
- Access Database Services, 60
- Access Services service application, 179, 181, 348
- ACL (access control list), 895
- actions
 - defined, 930
 - external data, 907–909
- Active Directory
 - claims-based authentication and, 70
 - installation considerations, 147–149
 - security groups, 818
 - server roles and, 141
 - user profiles, 503
- Active Directory Domain Services (AD DS), 918
- Active Directory Federation Services (ADFS), 270
- ActiveX controls, 145
- AD DS (Active Directory Domain Services), 918
- Add New form, 24
- Add-SPShellAdmin cmdlet, 206, 822
- ADFS (Active Directory Federation Services), 270
- Administration database, 103
- administrative tasks
 - application management, 253–289
 - backup and restore, 315–324
 - configuration wizards, 343
 - examples of, 234–241
 - General Application Settings, 333–343
 - monitoring, 299–315
 - performing basic, 232–233
 - publishing sites, 735–744
 - security, 325–330
 - system settings, 289–299
 - upgrade and migration, 331–333
- administrative user experience, 70–71
- Administrator Tasks Web Part, 251
- Advanced Search Box Web Part, 941
- Advanced Search Web Part, 640–644
- AIIM (Association for Information and Image Management), 441, 444
- alerts and notifications
 - delivering to mobile devices, 268
 - general settings, 265
 - setting up, 970–973
 - workflow process and, 731, 1103, 1106
- ALTER DATABASE statement, 106, 847
- alternate access mappings (AAM)
 - configuring, 276, 295
 - crawls and, 548
 - functionality, 780
 - UAG considerations, 979–981
- Analysis Services (Management Studio), 104
- angle brackets, 224
- anonymous authentication, 973, 1002
- ANSI/NISO Standard Z39.85-2007, 459
- anti-phrasing, 619
- antivirus settings, managing, 328
- API (Application Programming Interface)
 - additional information, 74
 - client object model, 74
 - column name considerations, 28
 - features and solutions, 74–75
 - fields and forms, 74
 - Web services, 74
- Application And Load Balance service application, 382

application management

- application management. *See also* Web applications
 - authentication providers, 270–271
 - blocked file types, 271, 329
 - databases, 287–289
 - general settings, 264–269
 - self-service site creation, 271, 283, 329
 - service applications, 283–287
 - site collections, 276–283
 - user permissions, 272
 - Web Application Ribbon, 256, 264–269
 - Web Part security, 273–274
- application pools
 - BCS considerations, 891
 - default, 90–91
 - functionality, 89–90, 261
 - IIS support, 89
 - memory considerations, 255
 - modifying, 369–371
 - resources needed, 89
 - security considerations, 90, 261
 - service applications and, 351
 - thread considerations, 89
 - Web applications and, 255
- application portal trunks, 981–988
- Application Programming Interface. *See* API (Application Programming Interface)
- application proxies, 352, 960
- Application Registry Service service application, 181
- application server roles
 - FAST Search Server support, 569
 - functionality, 82, 140, 388
 - hardware requirements, 142
 - identifying service locations, 356
 - planning topology, 361
 - software requirements, 144
 - User Profile Service and, 500
- Applicationhost.config file, 832
- Approval workflow, 76, 730–731, 1083–1084
- architectural framework
 - API considerations, 73–75
 - application pools, 89–91
 - ASP.NET 3.5 support, 63
 - capabilities, 75–78
 - database services, 62
 - deployment, 80–88
 - enterprise architecture, 58–59
 - management services, 70–72
 - master pages and, 64
 - operating system services, 62
 - personalization support, 64
 - planning for service applications, 357–361
 - planning topology, 361–364
 - provider framework and, 64
 - scaling service applications, 354–355
 - security services, 69–70
 - service architecture, 59–60
 - site model, 73
 - storage services, 65–68
 - topology services, 72
 - Web Parts Framework, 63
 - workflow services, 62
- Ask Me About Web Part, 967
- ASP.NET
 - field controls, 739
 - functionality, 63
 - SQL Server Reporting Services, 1001
 - workflow support, 1086
- assemblies, storing, 63
- asset libraries, 751
- Association for Information and Image Management (AIIM), 441, 444
- associations, service applications, 286
- asterisk (*), 219
- audiences, 499, 513–515
- auditing considerations
 - event audit trails, 1104
 - for information architecture, 467–468
 - generating audit reports, 428
 - in information management policies, 682
 - in retention policy, 409
 - in site collections, 800–801
- authentication
 - anonymous, 973, 1002
 - BCS support, 890–892
 - claims-based, 70, 257
 - classic-mode, 258
 - CredSSP, 206
 - defined, 890
 - digest, 1002
 - forms-based, 547
 - Kerberos, 259
 - mobile devices, 974
 - pluggable, 70
- authentication providers
 - changing for Web applications, 795–796
 - security considerations, 787–788
 - specifying, 270–271, 328
- authoring content. *See* content authoring
- authoritative pages, 594–596
- authorization, 894–895

B

- back tick (`), 582
- back-end servers, 608
- back-to-back firewalls, 976
- backup and restore. *See also* disaster recovery
 - as storage service, 67–68
 - catastrophic, 861–863
 - Central Administration support, 315–324, 850–851, 867–869
 - content databases, 264
 - farm, 67, 316–320, 866–870
 - granular backup, 322–324
 - role of, 827–830
 - site collections, 322, 863
 - SQL Server support, 870
 - storage considerations, 828
 - STSADM command support, 861–866
 - transaction log files, 128
 - troubleshooting, 852
 - types of backups, 828–829
 - upgrade preparations and, 1046–1047
 - Windows PowerShell support, 851–860, 866
- BACKUP DATABASE command, 845
- BACKUP LOG command (SQL Server), 128
- Backup-SPConfigurationDatabase cmdlet, 853, 856, 860
- Backup-SPFarm cmdlet, 853–855, 860
- Backup-SPSite cmdlet, 853, 857–858, 860
- barcodes, 410, 682, 785
- Basic Meeting Workspace template, 477
- Basic Search Center site template, 631
- BCS (Business Connectivity Services)
 - building composite solutions, 912–919
 - external data searches, 916–918
 - managing data connections, 896–912
 - Microsoft Office integration, 49
 - overview, 882–883
 - security options, 890–895
 - user profiles, 503
 - using BDC Model, 883–889
- BDC (Business Data Connectivity)
 - administration, 885
 - databases supported, 103
 - functionality, 60, 181, 348
 - modifying objects, 909–910
 - runtime considerations, 886
- BDC Model
 - creating, 900–902
 - creating external data actions, 907–909
 - importing, 903–905
 - modifying BDC objects, 909–910
 - using, 883–889
- BDC_Service_DB database, 103
- bdcmetaman tool, 887
- best bets, 628–629
- bit rate throttling, 756
- Blank Meeting Workspace template, 477
- Blank Site template, 476
- blank sites, 478
- BLOB cache, 753–756
- blocked file types, 271, 329
- Blog API, 265
- Blog template, 476
- blogs, defined, 493
- branding, 721, 732–733
- breadcrumb navigation, 25, 248
- Broadcast service, 964–965
- buffer cache, 112–113
- Business Connectivity Services. *See* BCS (Business Connectivity Services)
- Business Data Actions Web Part, 930
- Business Data Catalog Filter Web Part, 931
- Business Data Connectivity. *See* BDC (Business Data Connectivity)
- Business Data Item View Web Part, 967
- Business Data Item Web Part, 931
- Business Data Web Parts, 914, 930–932, 936
- business intelligence. *See also* PerformancePoint Services; SQL Server Reporting Services
 - decision making and, 989–990
 - key performance indicators, 991–993
- business scorecards, 992–993
- business taxonomy, 454

C

- cache profiles, 759–760
- caching. *See* content caching
- calculated columns, 31–32
- call centers, 443
- CALs (client access licenses), 137
- cascade deletes, 31
- case sensitivity
 - custom dictionaries, 530
 - Windows PowerShell cmdlets, 107
- catastrophic backups and restores, 861–863
- Categories Web Part, 940
- Central Administration. *See also* application management
 - accessing, 244

Central Administration user database

- administrative user experience, 70–71
- backup and restore support, 315–324, 850–851, 867–869
- breadcrumb navigation, 248
- Browse option, 249
- Collaboration tab, 475
- configuration wizards, 343
- creating BCS applications, 883
- disaster recovery and, 833, 836, 841–842
- Edit Page option, 251
- General Application Settings, 333–343
- large list throttling, 37
- monitoring, 299–315
- navigating home page, 246–252
- overview, 243–246
- Page option, 250–252
- security, 325–330
- service applications, 60
- Site Actions menu, 247–248
- Site Settings link, 11
- SQL Server Reporting Services and, 1012–1016
- system settings, 289–299
- topology services and, 72
- upgrade and migration, 331–333
- Central Administration user database, 102, 184–187
- certificate authority, 260
- Chart Web Part, 14
- checkpoints, defined, 127
- chkdsk command, 131
- chm file, 223
- claims-based authentication, 70, 257
- classic-mode authentication, 258
- Click To Edit feature, 12–13
- client access licenses (CALs), 137
- client computers, software requirements, 145–146
- client object model, 74
- cmdlets. *See also* specific cmdlets
 - administering content sources, 587–592
 - backup and restore support, 851–860
 - basic search capability, 574–577
 - case sensitivity, 107
 - configuring authoritative pages, 594–596
 - creating/managing search topology, 578–586
 - disaster recovery and, 844
 - enterprise search capability, 577
 - for SharePoint 2010, 231–232
 - for SharePoint Foundation 2010, 230
 - functionality, 214–215
 - metadata considerations, 596–601
 - parameters in, 221, 224–228
 - performance considerations, 219
 - properties and methods, 218–219
 - service application equivalents, 179
 - using verbs, 220–221
 - whatif parameter, 233
 - working with objects, 215–216, 228–229
- CodeActivity activity, 1095
- collaboration site templates, 475–476
- collaboration sites
 - defined, 8, 473–474
 - enabling features, 477–478
 - Enterprise Wikis, 490
 - functionality, 474–475
 - integrating, 484
 - meeting workspace site templates, 476–477
 - moving content from, 741–742
 - planning, 479–484
 - portal sites and, 486
 - rating content quality, 495
 - sharing knowledge, 491–496
 - site templates supported, 9
- Colleague Tracker Web Part, 967
- Collect Feedback workflow, 1084
- Collect Signatures workflow, 1084
- columns. *See also* list columns; site columns
 - calculated, 31–32
 - external data, 914–915
 - information, 29
 - lookup, 29–31
 - mobile view support, 967–969
- communities, social networking, 77, 491
- Company Portal user database, 102
- compiled workflows, 1080
- CompleteTask activity, 1095
- compliance officers, 400
- compliance requirements document, 399
- conditional formatting, 945
- configuration. *See* specific configuration wizards
- Connector Framework
 - FAST Search Server support, 571–572
 - indexing connectors, 571–572
 - search considerations, 531–532
- content approval
 - business processes for publishing, 729–730
 - functionality, 679
 - securing lists and, 819–820
- content authoring
 - digit media and asset libraries, 750–751
 - page authoring, 721
 - smart client, 746–748

- WCM support, 745
- Web Parts in publishing pages, 748–750
- Web-based, 745–746
- content caching
 - BLOB cache, 753–756
 - configuring, 752–767
 - implementation considerations, 752
 - object cache, 753, 764–767
 - overview, 753
 - page output cache, 753, 757–764
- content databases
 - associating with Web applications, 1057–1059
 - attaching to farms, 1056–1060
 - backing up, 264
 - detach, 1072
 - disaster recovery and, 846–848
 - indexing connectors, 571–572
 - Latin1_General_CI_AS_KS_WS option, 125
 - managing, 287–289
 - pre-creating, 124–126
 - read-only, 846–847, 1072
 - recovering data from, 324
 - recovery model, 127–129
 - reviewing status, 332
 - scaling service applications, 355
 - shrinking, 122–123
 - SQL Server Reporting Services and, 1005
 - unattached, 848
- content deployment
 - configuring, 771–777
 - considerations, 769–771
 - functionality, 76, 339–343
 - important concepts, 767–769
- content deployment jobs
 - creating, 774–776
 - overview, 768
 - Quick Deploy jobs, 776–777
- content deployment paths, 767, 772–774
- Content Editor Web Part, 14, 749, 938
- content management. *See* ECM (Enterprise Content Management); records management
- content managers, defined, 400
- Content Organizer feature
 - configuring, 686–687
 - creating rules, 420, 422
 - overview, 685
 - Record Center support, 417
 - setting up rules, 687–688
- Content Query Web Part
 - filtering support, 936
 - function, 938
 - functionality, 720, 748–750, 933–934
 - performance considerations, 924
 - publishing considerations, 925
- Content Rollup Web Parts, 929, 933–934
- content sites
 - defined, 9
 - site templates supported, 9
- content sources
 - adding, 916
 - administering, 587–592
 - crawl rules and, 532–535
 - creating and managing, 542–544, 551
 - managing crawls, 552
 - removing inadvertent exposure, 544
 - start address considerations, 538
- content types
 - configuring in site collections, 706–707
 - creating, 416–417
 - defined, 416, 450
 - digital asset management, 722
 - Enterprise Metadata Management, 694–696
 - information organization tools, 471
 - inheritance for, 417
 - publishing, 705–707
 - syndicating, 703–707
 - workflows by association, 1081, 1111–1112
- contextual promotions, 620–631
- conversion, document, 77, 335, 746–748
- Core Results Web Part, 654–655
- Crawl database, 103
- crawl databases, 564
- Crawl History Web Part, 554
- crawl logs, 554–557
- crawl reports, 558–559
- crawl rules
 - creating/managing, 544–547
 - defined, 338
 - indexes and, 547
 - managing, 537–538
- crawled properties, 596
- crawls
 - BCS considerations, 891
 - configuring, 541–550
 - content sources, 532–535
 - creating components, 564–565
 - enterprise search capability, 582–585
 - managing, 550–553
 - property considerations, 596, 917–918
 - reviewing, 554–561

Create New Web Application page

- scheduling, 552–553
- troubleshooting, 554–561
- Create New Web Application page
 - Application Pool section, 260–261
 - Customer Experience Improvement Program, 263
 - Database Name And Authentication section, 261
 - Failover Server section, 262
 - opening, 257
 - Public URL section, 260
 - Search Server section, 262
 - Security Configuration section, 258–260
 - selecting authentication method, 257–258
 - Service Application Connections section, 262
- CreateTask activity, 1095
- CredSSP authentication, 206
- CRM (Customer Relationship Management), 454
- CRUD functionality, 946
- Current User Web Part, 936
- custom dictionaries (search capability), 529–530, 657–659
- Customer Experience Improvement Program, 263
- Customer Relationship Management (CRM), 454
- Customize List section (Ribbon), 41

D

- Daniel, D. Ronald, 991
- Darwin Information Typing Architecture (DITA), 460
- DAS (Direct Attached Storage), 116
- dashboards
 - defined, 993
 - developer, 312–314, 924
 - overview, 993
- data cache, 112–113
- data integrity, verifying, 130
- data protection. *See* backup and restore; disaster recovery
- data retrieval services, 289
- Data View Web Part (DVWP), 936, 945–947
- database access account, 149
- database attach upgrades
 - associating Web applications/content databases, 1057–1059
 - attaching content databases to farms, 1056–1060
 - defined, 1036
 - preparing farm environment, 1053–1055
 - primary scenarios, 1038–1040
 - special cases, 1041
 - supported topologies, 1042
 - verifying environment, 1055
 - verifying permissions, 1056–1057
 - verifying successful completion, 1059–1060
- Database Console Command (DBCC), 123, 130
- Database Engine service (Management Studio), 104
- database files
 - default location, 116–117
 - modifying default location, 118
 - SQL Server support, 115
 - storage options, 116
- database mirroring, 871–872, 874
- database server role
 - FAST Search Server support, 569
 - functionality, 82, 140, 388
 - hardware requirements, 142
 - software requirements, 143
- Date Filter Web Part, 936
- DBCC (Database Console Command), 123, 130
- dbcreator privilege, 158
- DCMI (Dublin Core Metadata Initiative), 459–460
- Decision Meeting Workspace template, 477
- declarative workflows, 1080, 1082, 1101–1102, 1108
- delegation, 71, 507, 893
- DeleteTask activity, 1095
- deleting
 - list columns, 29
 - list items, 31
 - service applications, 284
 - user profile properties, 512
 - Web applications, 264
 - Web Parts, 923
- deployment. *See also* content deployment
 - development environments, 87
 - large farm, 86, 387, 394
 - medium farm, 85, 387, 392–393
 - My Site feature, 522–523
 - planning for workflow, 1098–1104
 - security considerations, 812
 - server roles and, 80–82
 - service applications, 365
 - small farm, 83–84, 387, 389–391
 - SQL Server Reporting Services, 1016–1017
 - testing environments, 87–88
 - Web browser considerations, 145
 - with FAST Search, 601–613
 - with Windows PowerShell scripts, 234–237
- developer dashboards, 312–314, 924
- development environment, 87
- diagnostic logging, 72, 307–308, 559–561
- dialog boxes, 14–15
- dictionaries, custom, 529–530, 657–659

- differential backups, 828
- digest authentication, 1002
- digital asset management
 - content types, 722
 - overview, 721–722, 750
 - user interface, 723
 - video streaming infrastructure, 722
- Direct Attached Storage (DAS), 116
- disaster recovery. *See also* backup and restore
 - defined, 826
 - documentation considerations, 834–837
 - education phase in planning, 831–834
 - full farm, 876–877
 - high-availability options, 871–876
 - importance of redundancy, 827
 - preparation, 837–838
 - tools supported, 838–848
- disk space, 110, 130–131
- Disposition Approval workflow, 1084
- distribution groups, 326, 783
- DITA (Darwin Information Typing Architecture), 460
- DNS records, 877
- DocBook taxonomy, 460
- Document Center template, 692
- document conversion, 77, 335, 746–748
- document libraries
 - adding documents, 676
 - adjusting settings, 677–678
 - check-in/check-out considerations, 679
 - Content Organizer feature, 685–688
 - creating, 675–676
 - Draft Item Security feature, 680, 822
 - metadata navigation/filtering, 683–684
 - mobile views and, 967
 - opening/saving documents, 682
 - overview, 674–675
 - versioning considerations, 678–679
- document management
 - Content Organizer, 685–688
 - disaster recovery planning, 834–837
 - Document Center template, 692
 - document libraries, 674–678, 682
 - document sets, 688–691
 - information management policies, 680–682
 - metadata default values, 691–692
 - metadata navigation/filtering, 683–684
 - overview, 674
 - versioning, 678–680
- Document Set Contents Web Part, 935
- Document Set Properties Web Part, 935

- document sets, 688–691, 935
- Document Workspace template, 476
- dollar sign (\$), 240
- double hopping process, 206
- Draft Item Security feature, 680, 822
- Drop Off Library Web Part, 928
- Dual Chinese Search Web Part, 941
- Dublin Core Metadata Initiative (DCMI), 459–460
- DVWP (Data View Web Part), 936, 945

E

- ECM (Enterprise Content Management), 438, 673–692
- ECT (external content type)
 - authorization and, 894
 - creating external data actions, 907–909
 - defined, 885–886
 - external data columns, 914
- edge firewalls, 975
- eDiscovery And Hold feature, 716–717, 786
- Edit form, 24
- Edit Page command, 13
- editing features
 - applying upgraded UI to sites, 18
 - Click To Edit feature, 12
 - creating rich themes, 14–17
 - dialog boxes, 14–15
 - incorporating rich media support, 18–19
 - rich text editing, 13, 21–24
 - SharePoint Ribbon, 13
- e-mail
 - managing distribution groups, 326
 - modifying user details, 265
 - outgoing e-mail, 267
 - searching messages, 494
 - setting up SMS alerts, 970–973
 - system settings, 292–295
- Enable-PSRemoting cmdlet, 204
- Enable-WSManCredSSP cmdlet, 207
- Enforce Unique Values option, 28
- enterprise architecture, 58–59
- Enterprise Content Management (ECM), 438, 673–692
- Enterprise Metadata Management. *See* Managed Metadata Service
- enterprise search. *See also* search capability
 - administering content sources, 587–592
 - configuring authoritative pages, 594–596
 - creating/managing topology, 578–586
 - metadata considerations, 596–601
 - overview, 526–528

- SharePoint Foundation and, 578
- Windows PowerShell and, 577
- Enterprise Search Center site template, 631
- Enterprise Wiki site template, 490, 495, 727
- Enter-PSsession cmdlet, 209
- entity extraction, 660–661
- event audit trails, 1104
- Excel Services service application
 - controlling settings, 962
 - functionality, 60, 181, 348
- Excel Web Access Web Part, 949
- Exit-PSsession cmdlet, 209
- expertise tagging, 494
- exporting
 - content deployment considerations, 770
 - lists, 323
 - sites, 323, 864
 - subsites, 864
 - trust certificates, 376–378
- Export-PSsession cmdlet, 209
- Export-SPWeb cmdlet, 853, 858–860
- Extensible Markup Language (XML), 289, 887, 945
- Extensible Metadata Platform (XMP), 461
- Extensible Object Markup Language (XOML), 1080
- external content type. *See* ECT (external content type)
- external data columns, 914–915
- external lists, 912–913
- external system throttling, 910–912

F

- failover clustering, 875
- Farm Configuration user database, 102
- Farm Configuration Wizard
 - application pools and, 90
 - creating BCS applications, 883
 - enterprise searches, 577–578
 - Office Web Application support, 959
 - running, 180–184, 343
 - search service applications and, 538
- farm management
 - attaching content databases, 1056–1060
 - backup and restore, 67, 316–320, 866–870, 1046
 - configuring mobile accounts, 971–972
 - configuring privacy options, 298
 - content deployment and, 770
 - converting farm license type, 331
 - disaster recovery and, 876–877
 - farm-wide search settings, 536
 - history of SharePoint farms, 346–347
 - identifying service locations, 356–364
 - identifying workflow roles, 1099
 - managing, 295–299
 - managing administrators group, 326, 541, 782–783
 - planning topology architecture, 361–364
 - publishing service applications, 376–386
 - scaling considerations, 354, 388–392
 - search administration, 337
 - security considerations, 782–785
 - Send To option, 423–424
 - Services Federation, 347–350
 - system settings, 290–291
 - transferring customizations, 1055
 - upgrade considerations, 1053–1055
 - workflows and, 1083
- FAST Query Language (FQL), 667
- FAST Search Center site template, 632
- FAST Search Server for SharePoint
 - architecture and topology, 568–570
 - comparing search features, 526–528
 - customizing search pages, 635–655
 - deploying with Windows PowerShell, 601–613
 - features supported, 618
 - functionality, 138, 566–568, 616–618
 - improving collection quality, 669–671
 - improving search quality, 656–668
 - improving user experience, 619–635
 - indexing connectors, 571–572
 - plug-ins supported, 567–568
 - security considerations, 605
- FASTSEARCH environment variable, 605
- FBA (forms-based authentication), 547
- Federated Results Web Part, 941
- federation, 59, 78
- Federation Gateways, 70
- field controls, 739
- file plan (records management)
 - creating, 419
 - functionality, 400–402, 718–719
 - generating report, 426–427
 - tracking record classifications, 416
- File Transfer Protocol (FTP), 254
- file types
 - blocked, 271, 329
 - caching considerations, 754
 - iFilters and, 531, 550
 - managing, 550
 - mobile device support, 959
- Filter Actions Web Part, 936

filtering

- Filter Web Parts, 936–937, 948–949
- list data, 33
- metadata, 683–684
- search information, 622
- XRANK method and, 667–668

findability (information architecture)

- developing, 469
- overview, 440–444, 446–449, 621
- tool development, 461–462

firewalls

- back-to-back, 976
- configuring access settings, 299
- disaster recovery planning, 832
- edge, 975
- e-mail and, 267, 293
- mobile devices and, 974–978
- multi-homed, 976
- UAG considerations, 979

folders, WCM support, 721

Forefront UAG. *See* UAG (Unified Access Gateway)

formatting, conditional, 945

forms services, 78, 336

Forms Web Parts, 937–938

forms-based authentication (FBA), 547

FQL (FAST Query Language), 667

FTP (File Transfer Protocol), 254

full backups, 829

full recovery model, 128–129

FullTextSQLQuery class, 623

G

GAAP (generally accepted accounting principles), 459

GAC (Global Assembly Cache), 63, 1080

generally accepted accounting principles (GAAP), 459

Get-Command cmdlet, 214, 220, 606

Get-Eventlog cmdlet, 235

Get-ExecutionPolicy cmdlet, 201

Get-Help cmdlet, 214, 221–223, 230

Get-Member cmdlet, 584, 588

Get-RemoteSPSite cmdlet, 209

Get-Service cmdlet, 203, 610

Get-SPBusinessDataCatalogThrottleConfig cmdlet, 911

Get-SPContentDatabase cmdlet, 214, 226

Get-SPDiagnosticConfig cmdlet, 235

Get-SPEnterpriseSearchCrawlContentSource cmdlet, 587–590, 592

Get-SPEnterpriseSearchCrawlTopology cmdlet, 583–584, 587

Get-SPEnterpriseSearchExtendedConnectorProperty cmdlet, 610

Get-SPEnterpriseSearchExtendedQueryProperty cmdlet, 612–613

Get-SPEnterpriseSearchIndexPartition cmdlet, 586

Get-SPEnterpriseSearchMetadataCrawledProperty cmdlet, 597

Get-SPEnterpriseSearchMetadataManagedProperty cmdlet, 623

Get-SPEnterpriseSearchMetadataMapping cmdlet, 601

Get-SPEnterpriseSearchQueryAuthority cmdlet, 595

Get-SPEnterpriseSearchQueryTopology cmdlet, 583, 587

Get-SPEnterpriseSearchServiceApplication cmdlet, 578, 587, 592

Get-SPEnterpriseSearchServiceInstance cmdlet, 582, 610

Get-SPSearchService cmdlet, 574

Get-SPSearchServiceInstance cmdlet, 574, 576

Get-SPServiceApplicationPool cmdlet, 580

Get-SPServiceApplicationProxyGroup cmdlet, 582, 911

Get-SPServiceInstance cmdlet, 207

Get-SPSite cmdlet, 219, 224, 228

Get-SPSiteAdministration cmdlet, 228

Get-SPWebApplication cmdlet, 592

Get-SPWebApplicationhttp cmdlet, 1075

Get-SPWebTemplate cmdlet, 228

Global Assembly Cache (GAC), 63, 1080

Globally Unique Identifier (GUID), 184

governance

- information architecture and, 456
- information organization project, 470
- managing requirements to plan, 457–458
- records management and, 710

granular backups, 68, 829

granular restores, 830

Group Work Site template, 476

groups

- permission levels, 816–818
- security, 818

GUID (Globally Unique Identifier), 184

H

hardware requirements

- disaster recovery planning, 832, 837
- for installing and configuring, 109–111, 141–142

health analyzers, 72, 299–302

Health Insurance Portability and Accountability Act (HIPAA), 397

high-availability options, 871–876

Host Distribution Rule, 535, 548

HTML Form Web Part, 14, 749, 937
HTML Viewer Web Part, 937
HTTP.SYS (listener), 1001–1002
hub site collections, 77
human workflows, 78, 1078
hyphen (-), 221

I

IBM Tivoli Directory Server (ITDS), 918
identity delegation, 893
IDisposable Interface, 228
iFilters, 531, 550
iFrames, 939
IIS (Internet Information Server)
 application pools and, 89, 255
 disaster recovery planning, 832, 836
 managing systems remotely, 203
 managing Web applications, 254
 operating system services and, 62
 service applications and, 350
 SQL Server Reporting Services and, 1002
 time-based execution, 1082
Image Viewer Web Part, 749, 967
Import Reusable Workflow project template, 1094
importing
 BDC Model, 903–905
 content deployment considerations, 770
 sites, 864–865
 subsites, 864–865
 trust certificates, 379–381
Import-PSession cmdlet, 209
Import-SPWeb cmdlet, 853, 859–860
Inbox Web Part, 938
indexes
 automatic column, 684
 Connector Framework support, 571–572
 crawl rules and, 547
 FAST Search Server support, 571–572
 list columns, 32–33
 number sequences, 530
 partitioning, 534, 565
 resetting, 550
 search capability and, 78
 site/list content, 802
 specifying locations, 585
InfoPath 2010
 Design Form Page, 43
 forms services support, 336
 modifying list input forms, 42–45
 upgraded form templates, 1075
 workflow support, 1086
InfoPath Form Web Part, 748, 938, 950
information architecture. *See also* findability (information architecture); putability (information architecture)
 assessing, 463–465
 audit and analysis, 467–468
 building, 451–462
 business requirements development, 465
 call center example, 443
 developing, 437–444
 governance and requirements, 454–457, 470
 information organization project, 462–470
 information organization tools, 471–472
 maintaining, 462
 Managed Metadata Services and, 449–450
 opportunity costs, 442–444
 project charter, 466
 technical requirements, 466
 usability and tool development, 461–462
 value of information, 438–439
information columns, 29
information management policies. *See also* records management
 applying, 422
 auditing, 682
 barcodes, 682, 785
 configuring, 404–414, 784–785
 document management and, 680–682
 generating usage reports, 411–413
 implementing, 404–414
 In Place Records Management feature, 433
 labels, 682, 784
 overview, 402–403
 planning document policies, 403
 policy metadata, 404
 retention, 681, 785
 reviewing usage reports, 315
 security considerations, 330
 viewing usage reports, 413–414
information organization. *See* information architecture
information repository, 65–66
information rights management (IRM), 547, 784
information workers, 400, 1099
In-Place Records Management feature
 configuring, 430–433
 implementing, 429
 overview, 713–715

- in-place upgrades
 - configuring, 1066–1070
 - defined, 1036
 - hybrid upgrade approach, 1072
 - installing, 1062–1065
 - primary scenarios, 1038–1040
 - process overview, 1061–1062
 - restrictions, 1036, 1042–1044, 1061
 - SharePoint Portal Server and, 1073–1074
- Integration Services (Management Studio), 104
- Interest Tracker Web Part, 967
- Internet Information Server. *See* IIS (Internet Information Server)
- Invoke-Command cmdlet, 208
- IPSV taxonomy, 461
- IRM (information rights management), 547, 784
- ISO Standard 15836
 - 2009, 459
- ITDS (IBM Tivoli Directory Server), 918
- iView Web Part, 14

K

- Kerberos authentication, 259
- key performance indicators (KPIs), 991–993
- Knowledge Network, 494
- knowledge sharing. *See* sharing knowledge
- KPIs (key performance indicators), 991–993

L

- labels
 - for information management policies, 682, 784
 - for retention policies, 410–411
- language detection (search capability), 529, 631
- large farm deployment, 86, 387, 394
- LDAP (Lightweight Directory Access Protocol), 70, 503, 918
- lemmatization, 659–660
- libraries. *See* lists and libraries
- Lightning Conductor Web Part, 936
- Lightweight Directory Access Protocol (LDAP), 70, 503, 918
- linguistic processing, 656
- list columns
 - creating, 25, 27–28
 - defined, 25
 - deleting, 29
 - indexing, 32–33
 - sorting and filtering data, 33

- types supported, 29–32
- validating, 28
- viewing, 28
- list input forms
 - editing, 41–49
 - modifying form Web Parts, 41
 - modifying with InfoPath, 42–45
 - modifying with SharePoint Designer, 46–49, 74
 - overview, 20–24
- list items
 - defined, 25
 - deleting, 31
- List Settings option (Ribbon), 28
- list templates
 - default, 26
 - functionality, 25–26
 - site model and, 73
- List View Web Part, 929, 936–937, 967
- list views
 - changing, 35
 - creating, 25, 34–35
 - functionality, 34
 - mobile access, 35
 - personalization support, 64
- lists and libraries. *See also* document libraries
 - creating, 26
 - declaring records in libraries, 714
 - defined, 20
 - exporting, 323
 - external lists, 912–913
 - identifying workflow roles, 1100
 - In Place Records Management feature, 430–433
 - indexing content, 802
 - information organization tools, 471
 - large list support, 36–39
 - list enhancements, 20
 - list input forms, 20–24, 41–49
 - list templates, 25–26
 - managing list columns, 27–36
 - mobile view support, 967–968
 - monitoring workflow status, 1114–1115
 - provisioning, 71
 - rating content quality, 495
 - security considerations, 819–822
 - site columns and, 39–40
 - site model and, 73
 - team collaboration features, 477
 - user permissions, 272
 - workflows by association, 1081–1082, 1109–1111
- Lists And Libraries Web Parts, 928

- log shipping, 872–874
- Logging database, 102
- LogToHistoryList activity, 1095
- long tail concept, 446
- lookup columns, 29–31
- Lotus Notes Connector, 181, 532, 572

M

- managed accounts, configuring, 327
- managed keywords, 625–627
- Managed Metadata Service
 - configuration settings, 693–694
 - content type syndication, 703–707
 - content types/columns, 694–696
 - creating content types, 417
 - databases supported, 103
 - functionality, 60, 77, 181, 348
 - information organization tools, 471
 - managed metadata fields, 697
 - overview, 692–694
 - post-upgrade configurations, 1074
 - putability and, 449–450
 - scaling service applications, 355
 - setting up, 704–705
 - social tagging and, 493
 - term store and term sets, 698–703
 - User Profile Service and, 501
- managed paths
 - functionality, 269
 - organizing tools and, 471
 - planning for collaboration, 480–482
- managed properties, 596
- management services
 - administrative user experience, 70–71
 - delegation, 71
 - elements supported, 70
 - metadata, 77
 - monitoring, 72
 - multi-tenancy, 72
 - provisioning, 71
 - Windows PowerShell, 72
- Management Studio (SQL Server), 104–105
- master pages, 64, 265, 739
- Master system database, 101
- max server memory option, 114
- Measure-Command cmdlet, 215
- Media Web Part, 14, 748
- medium farm deployment, 85, 387, 392–393
- meeting workspace site templates, 476–477
- memory
 - application pools and, 255
 - default settings, 113
 - setting options manually, 114
 - SQL Server optimization, 112–115
 - SQL Server Reporting Services and, 1003
 - SQL Server requirements, 110
 - Windows PowerShell and, 228–230
- metadata. *See also* Managed Metadata Service
 - as storage service, 66
 - BCD Model, 888–889
 - customizing search results, 654–655
 - defined, 66, 596
 - filtering, 683–684
 - information management policies, 404
 - location-based default values, 691–692
 - maintaining categories and properties, 596–601
 - managed services, 60, 77
 - navigating, 683–684
 - records management and, 422
 - user context, 621–625
- metadata store, 885
- methods, defined, 218
- Microsoft Forefront UAG. *See* UAG (Unified Access Gateway)
- Microsoft Groove, 49, 76
- Microsoft Office
 - integration considerations, 49–55
 - software requirements, 146
 - using media from applications, 751
 - Web Part support, 949–950
- Microsoft Office Mobile, 954–955
- Microsoft Office Web Applications. *See* Office Web Applications
- Microsoft Operations Manager Extension for SharePoint 2010, 670
- Microsoft PerformancePoint Server, 78
- Microsoft PowerPoint, 15–16, 964–965
- Microsoft Silverlight
 - software requirements, 146
 - Web Part support, 14, 749, 950
- Microsoft SQL Server. *See* SQL Server (Microsoft)
- Microsoft Visio 2010, 1085–1088, 1091–1092
- Microsoft Word, 53, 67
- Microsoft.Windows.SharePoint.Workflow namespace, 1096
- migration. *See* upgrade and migration
- min server memory option, 114
- MMC snap-ins, 195, 243

- mobile devices. *See also* SharePoint Mobile
 - configuring e-mail settings, 295
 - configuring mobile accounts, 971–973
 - delivering alerts to, 268
 - enabling access to portal trunks, 988
 - exploring access scenarios, 973–974
 - file types supported, 959
 - list views, 35
 - overview, 953–954
 - previewing access options, 969–970
 - setting up SMS alerts, 970–973
 - usage considerations, 956–957
- mobile search, 966
- mobile views, 967–969
- Model system database
 - Auto Shrink option, 122
 - Autogrowth setting, 120–122
 - default settings, 119
 - functionality, 102
 - inheriting settings, 118
 - Initial Size setting, 120
 - modifying settings, 120–122
 - recovery model, 120
 - shrinking content databases, 122–123
- monitoring
 - buffer cache, 113
 - health analyzers and, 299–302
 - management services support, 72
 - PerformancePoint Services, 1019–1020
 - reporting, 306–315
 - SQL I/O, 132–133
 - timer jobs, 304–306
 - workflows, 1114–1116
- Moreville, Peter, 621
- Move-SPProfileManagedMetadataProperty cmdlet, 1074
- MsdB system database, 101
- MSSCrawlQueue table, 533
- MSSPropagationErrors table, 534
- MSSPropagationLog table, 534
- MSSPropagationTaskCompletions table, 534
- MSSPropagationTasks table, 534
- MSSQLServer service, 98
- multi-homed firewalls, 976
- Multipage Meeting Workspace template, 477
- My Content (My Site), 492
- My Newsfeed (My Site), 492
- My Profile (My Site), 492
- My Site feature
 - configuring settings, 519–520

- functionality, 77, 499
- global deployment, 522–523
- hosting considerations, 499
- post-upgrade configurations, 1075
- setting up, 519–521
- sharing knowledge via, 491–492
- social tagging and, 493
- User Profile Service and, 473, 491

N

- named instances, 99
- naming conventions, custom dictionaries, 530
- NAS (Network Attached Storage), 116
- NASCIO (National Association of State Chief Information Officers), 826
- navigation
 - breadcrumb, 25, 248
 - Central Administration home page, 246–252
 - metadata, 683–684
 - publishing sites, 736–738
 - security trimming, 801
 - site model and, 73
 - Web Part support, 940
- Network Attached Storage (NAS), 116
- New Site command, 12
- New-PSSession cmdlet, 207
- NewsML taxonomy, 461
- New-SPAccessServiceApplication cmdlet, 179
- New-SPBusinessDataCatalogServiceApplication cmdlet, 884
- New-SPEnterpriseSearchCrawlComponent cmdlet, 584–585
- New-SPEnterpriseSearchCrawlContentSource cmdlet, 587, 590–591
- New-SPEnterpriseSearchCrawlTopology cmdlet, 585
- New-SPEnterpriseSearchExtendedConnectorProperty cmdlet, 610–611
- New-SPEnterpriseSearchMetadataManagedProperty cmdlet, 598
- New-SPEnterpriseSearchQueryAuthority cmdlet, 595
- New-SPEnterpriseSearchQueryComponent cmdlet, 586
- New-SPEnterpriseSearchQueryDemoted cmdlet, 596
- New-SPEnterpriseSearchQueryTopology cmdlet, 586
- New-SPEnterpriseSearchServiceApplication cmdlet, 581, 610, 613
- New-SPEnterpriseSearchServiceApplicationProxy cmdlet, 581
- New-SPManagedAccount cmdlet, 580

New-SPPerformancePointServiceApplication cmdlet, 179
New-SPProfileServiceApplication cmdlet, 179
New-SPServiceApplicationPool cmdlet, 580
New-SPStateServiceApplication cmdlet, 179
New-SPVisioServiceApplication cmdlet, 179
normalization, 657
Note Board, defined, 492
notifications. *See* alerts and notifications
NTFS allocation unit size, 131–132
NTLM (NT LAN Manager), 259, 270

O

object cache, 753, 764–767
Object Linking and Embedding, Database (OLEDB), 289
Office 2010 Mobile Service, 971
Office Communication Server, 77
Office Communicator, 77
Office Web Applications
 activating services, 960
 activating site collection features, 961–962
 checking services status, 959–960
 controlling service-specific settings, 962–964
 creating application proxies, 960
 file types supported, 959
 installing, 959
 overview, 349, 957–959
OLEDB (Object Linking and Embedding, Database), 289
OnTaskChanged activity, 1095
OnWorkflowActivated activity, 1095
OnWorkflowModified activity, 1095
Open permission, 808
OpenDocumentFormat taxonomy, 461
OpenSearch model, 534
operating systems
 disaster recovery planning and, 831, 835
 service functionality, 62
 software requirements, 143
operational taxonomies, 459–461
operators, Windows PowerShell, 227
opportunity costs, 442–444
optimizing SQL Server
 content database recovery model, 127–129
 database files, 115–117
 disk storage options, 130–131
 memory settings, 112–115
 Model database settings, 118–123
 monitoring SQL I/O, 132–133
 NTFS allocation unit size, 131–132
 pre-creating content databases, 124–126
 Tempdb database, 126–127
 verifying data integrity, 130
organization profiles, 498, 518–519
organizing information. *See* information architecture
Outlook Social Connector, 494, 496
Outlook Web Access (OWA), 938–939

P

page authoring, 721
Page Filter Web Part, 936
page layouts, 739
page model, 739–740
page output cache, 753, 757–764
Page Viewer Web Part, 749, 938
parameters in cmdlets, 221, 224–228
passphrases, 169
passwords, changing settings, 328
people search, 77, 495, 663
performance analysis reports, 1118–1119
PerformancePoint Dashboard Designer, 1032–1033
PerformancePoint Filter Web Part, 948
PerformancePoint Services
 adding applications, 1027
 architectural overview, 1020–1021
 cmdlet equivalent, 179
 creating applications, 1022–1024
 creating secure store service, 1024–1025
 function, 1018
 functionality, 60, 181, 348
 installing/configuring, 1022–1029
 migrating content, 1032–1033
 new features, 1018–1020
 trusted content locations, 1028
 trusted data source locations, 1028
 unattended service accounts, 1025–1026
 upgrading, 1029–1032
 Web Part support, 947–948
PerformancePoint Web Parts, 947–948
permission dependencies, 808–812
permission levels (roles)
 defined, 69, 806
 group considerations, 816–818
 inheriting, 813–815
 overview, 806–808
 Web applications and, 274, 793
permissions (rights)
 application management, 272
 BCS considerations, 895

- configuring for My Site feature, 521
- configuring for service applications, 286
- defined, 69
- dependencies among, 808–812
- overview, 803–806
- removing, 273
- self-service site creation, 792
- service applications, 823
- verifying for SQL Server, 1056–1057
- WCM considerations, 735–736
- workflow considerations, 1102
- personal views, 64
- Picture Library Slideshow Web Part, 749, 938
- pipe (|), 219, 582
- pluggable authentication, 70
- Pluggable Workflow Service, 1082
- plus sign (+), 1002
- policies for Web applications, 274–276
- portal sites
 - collaboration sites and, 486
 - functionality, 473
 - possible scenarios, 486–489
 - project implementation scenario, 487–489
 - publishing, 487
 - publishing features, 489–490
 - records management, 487
 - sharing knowledge, 491–496
- portal trunks
 - creating, 981–984
 - publishing in UAG, 984–988
- PowerPoint (Microsoft), 15–16, 964–965
- PowerPoint Broadcast Slide Show feature
 - adjusting service settings, 965
 - overview, 964
 - starting broadcasts, 964
- PowerPoint service application, 348, 963
- PowerShell (Windows). *See* Windows PowerShell
- Prerequisiteinstaller.exe tool
 - command-line installations, 149–150
 - GUI installation, 151–152
- pre-upgrade checker tool, 1047–1051
- primary data file, 115
- privacy options, configuring, 298
- procedure cache, 112–113
- processor, SQL Server requirements, 110
- product keys, enabling features, 331
- Profile database, 103, 501
- profile pages
 - configuring, 898–900
 - creating, 905
- Profile Synchronization Service, 498
- project implementation scenarios, 462–470, 487–489
- project templates, 1093
- properties
 - configuring extended, 611
 - configuring for service applications, 286
 - crawled, 596, 917–918
 - defined, 218
 - managed, 596
 - Search service application, 541
 - user profile, 509–512
- Property database, 103
- property databases, 565
- protocol handlers, 531–533
- provider framework, 58, 64
- provisioning
 - custom applications, 484
 - management services support, 71
 - planning for collaboration, 482–484
 - queries, 586–587
 - Search service applications, 580
 - service applications, 179
- Proxy cmdlet, 207
- proxy groups
 - service applications, 352–353, 371–372
 - Web applications, 366–367
- Psconfig.exe tool, 177–178, 234
- Psconfigui.exe tool, 158
- PSSnapin cmdlet, 207
- public host names, 979
- public URL, 260–261
- publishing content types, 705–707
- Publishing Portal site template, 727
- publishing portal sites, 487, 489–490
- publishing portal trunks, 984–988
- publishing Search service application, 540–541
- publishing service applications
 - exchanging trust certificates, 376–383
 - explicitly connecting, 385–386
 - explicitly publishing, 384–385
 - functionality, 353, 373–375
- publishing sites
 - administering, 735–744
 - branding and, 732–733
 - choosing site types, 727
 - configuring content caching, 752–767
 - defined, 9
 - enabling features, 728
 - navigating, 736–738
 - page model, 739–740

publishing Web content

- pages versus subsites, 736–737
- publishing portal features, 487, 489–490
- rating content quality, 495
- rich media support, 751
- security and permissions, 735–736
- SharePoint Designer settings, 740
- site templates supported, 9, 727
- Variation feature, 733
- WCM support, 726–734
- Web Analytics support, 742–744
- Web Part support, 748–750
- publishing Web content, 745–752
- publishing workflows, 729–731
- putability (information architecture)
 - defined, 437, 439–440
 - developing, 469
 - Managed Metadata Services and, 449–450
 - operational taxonomies and, 459–461
 - tool development, 461–462

Q

- QRProxyService.exe.config file, 610, 613
- queries. *See* search capability
- Query Analyzer, 105
- Query Federation, 534
- Query Processor service, 534
- Query String (URL) Filter Web Part, 937
- Quick Deploy jobs, 776–777
- Quick Steps, creating, 46–47
- quota templates, 265, 280

R

- RAID arrays, 117, 130
- rank profile customizations, 663–667
- rating content quality, 495, 720
- RBS (Remote BLOB Storage), 65
- RDL (Report Definition Language), 998–999
- read-only databases, 846–847, 1072
- Recent Activities Web Part, 967
- records archive, 709
- Records Center
 - configuring, 416
 - Content Organizer and, 417, 420, 422
 - creating, 417–421
 - creating content types, 416–417
 - creating records libraries, 421
 - defining metadata, 422
 - generating audit reports, 428
 - generating file plan report, 426–427
 - Holds option, 424
 - implementing, 416
 - information organization tools, 471
 - managing content types, 416–417
 - records management and, 709
 - Send To function, 423–424
- Records Center site template, 711–712
- Records Center Web Service Submitters group, 418
- records libraries, 421
- records management. *See also* information management policies; Records Center
 - common tasks, 419
 - compliance requirements document, 399
 - document compliance and, 715
 - eDiscovery And Hold feature, 716–717, 786
 - features supported, 710–711
 - file plan, 400–402
 - in-place, 429–433, 713–715
 - overview, 397–399, 709
 - placing hold on records, 424
 - portal sites and, 487
 - Records Center site template, 711–712
 - retention considerations, 717–719
 - roles and responsibilities, 399–400
 - scalability, 719
 - site/list level declaration settings, 713–714
 - upfront preparation, 709–710
- records managers, defined, 400, 709
- recovery models
 - content databases, 127–129
 - full recovery model, 128–129
 - Model databases, 120
 - simple recovery model, 129
- Recycle Bin, 68, 266, 839–841
- Refinement Panel Web Part, 942
- regex expressions, 545–546, 662–663
- registry keys, 202
- Related Queries Web Part, 942
- Relevant Documents Web Part, 749
- Remote BLOB Storage (RBS), 65
- Remote Server Administration Tools (RSAT), 202
- Remove-PSSession cmdlet, 209
- Remove-SPEnterpriseSearchCrawlContentSource cmdlet, 587, 591
- Remove-SPEnterpriseSearchMetadataManagedProper ty cmdlet, 601
- Remove-SPEnterpriseSearchQueryAuthority cmdlet, 595
- Remove-SPSite cmdlet, 233–234
- Report Builder application, 1000

report catalog, 1005
 Report Definition Language (RDL), 998–999
 Report Server Execution Log, 1004
 Report Server Service Trace Log, 1004
 Report Viewer Web Part, 1001, 1008
 Reporting database, 103
 Reporting Services. *See* SQL Server Reporting Services
 reporting, monitoring, 306–315
 Reports Web Part, 947
 ReportServer database (RSDB), 1005
 ReportServerTempDB database (RSTempDB), 1005
 repository, information, 65–66
 request-based site creation, 483–484
 resource throttling. *See* throttling
 Resources Web Part, 244–245
 REST services, 945, 947
 restore. *See* backup and restore
 Restore-SPFarm cmdlet, 853–855, 857, 860
 Restore-SPSite cmdlet, 853, 857–858, 860
 restricted deletes, 31
 retention policies

- auditing feature, 409
- defining, 406–411
- document barcodes, 410
- information management policies, 681, 785
- labeling considerations, 410–411
- multiple stages in, 408
- records management, 717–719
- setting expiration dates, 407

 reverse proxies, 276, 974
 RFC 5013, 459
 Ribbon

- Central Administration support, 250
- creating Quick Steps, 46–47
- Customize List section, 41
- depicted, 13, 26, 52
- Editing Tools tab, 21, 25
- enhancements, 10
- Items tab, 26
- Link Tools tab, 23
- List Settings option, 28
- Page tab, 13
- Picture Tools tab, 22
- Sync tab, 52
- Table Tools Design tab, 23
- Table Tools Layout tab, 23

 rich media

- defined, 18
- incorporating support, 18–19
- within publishing sites, 751

rich text editing, 13, 21–24
 rights. *See* permissions (rights)
 rights trimming, 69
 Rockart, Jack F., 991
 roles (permission levels)

- defined, 69
- identifying for workflows, 1098–1100
- Web applications and, 274, 793

 rollback strategy, testing, 1046
 root authority, 379–380
 RSDS files, 999
 RSS feeds, 265, 471
 RSS Viewer Web Part, 933, 967

S

Safe Controls List feature, 63
 Safe Mode Parser feature, 63
 SAN (Storage Area Network), 116, 832
 Sarbanes-Oxley Act (SOX), 398
 scaling considerations

- FAST Search Server, 570
- search service topology, 562–563
- server groups, 388–392
- service applications, 354–355

 schedule types, 588
 scheduling

- content deployment, 770
- crawls, 552–553

 Scorecard Web Part, 947
 scorecards, business, 992–993
 SCORM taxonomy, 460
 scripts, Windows PowerShell, 593–594
 Search Action Links Web Part, 666, 942
 Search Best Bets Web Part, 943
 Search Box Web Part, 635–640
 search capability. *See also* FAST Search Server for SharePoint

- architectural overview, 528–529
- basic searches, 574–577
- benefits, 615–616
- components and processes, 532–535
- configuring applications, 540–541
- configuring crawls, 541–550
- crawler impact rules, 537–538
- creating applications, 538–540
- creating search rules, 661–663
- custom dictionaries, 529–530, 657–659
- customizing search pages, 635–655
- diagnosing content issues, 670–671

Search Center

- entity extraction, 660–661
- external data searches, 916–918
- farm-wide search settings, 536
- features supported, 618
- iFilters, 531, 550
- improving collection quality, 669–671
- improving search quality, 656–668
- improving user experience, 619–635
- language detection, 529, 631
- lemmatization, 659–660
- linguistic processing, 656
- managing crawls, 550–553
- managing search service topology, 561–566
- mobile devices, 966
- normalization, 657
- products overview, 526–528
- rank profile, 663–667
- refinement panel, 645–649
- reviewing crawls, 554–561
- search support staff, 536
- spelling and pronunciation, 657–659
- tokenization, 657
- tools supported, 529–532
- troubleshooting crawls, 554–561
- word breakers, 529
- XRANK method, 667–668
- Search Center
 - creating search pages, 632–635
 - customizing, 631–632
 - customizing search pages, 635–655
 - customizing search results metadata, 654–655
 - improving collection quality, 669–671
 - improving search quality, 656–669
- Search Core Results Web Part, 942
- Search Paging Web Part, 943
- Search Query and Site Settings service, 534
- Search Server, 526–528, 833
- Search service applications
 - administering content sources, 587–592
 - configuring, 540–541
 - configuring authoritative pages, 594–596
 - configuring crawls, 541–550
 - creating/managing topology, 578–586
 - databases supported, 103
 - disaster recovery planning, 837
 - Farm Configuration Wizard and, 538
 - FAST Search Server support, 567
 - functionality, 60, 77, 182, 348
 - general application settings, 337–338
 - managing topology, 561–566
 - metadata considerations, 596–601
 - modifying properties, 541
 - planning topology, 362
 - provisioning, 580
 - publishing, 540–541
 - User Profile Service and, 501
- search sites, 9–10
- Search Statistics Web Part, 943
- Search Web Parts, 14, 940–943
- Searchextensions feature, 603
- SEC (Securities and Exchange Commission), 459
- secondary data files, 116
- Secure Store service application
 - creating, 1024–1025
 - databases supported, 103
 - functionality, 60, 182, 348
- Secure Token Service (STS), 891
- Securities and Exchange Commission (SEC), 459
- security. *See also* firewalls; permissions (rights)
 - application pools, 90, 261
 - ASP.NET 3.5 support, 63
 - BCS considerations, 890–895
 - claims-based authentication, 70
 - content deployment and, 769
 - Draft Item Security feature, 680, 822
 - farm-level, 782–785
 - FAST Search Server, 605
 - general settings, 326–329
 - information policy and, 330
 - limiting PowerShell commands, 207
 - list considerations, 819–822
 - managing administrators, 326
 - managing distribution groups, 326
 - overview, 779–782
 - PerformancePoint Services, 1019
 - pluggable authentication, 70
 - registry keys, 202
 - removing inadvertent exposure, 544
 - rights and roles, 69, 272
 - rights trimming, 69
 - site collections, 800–801
 - site considerations, 10, 801–818
 - upgrade considerations, 1075
 - WCM considerations, 735–736
 - Web application user policy, 326
 - Web applications, 786–799
 - Web page validation, 265
 - Web Parts and, 273–274, 329, 796–797
 - Windows PowerShell, 822
 - workflow considerations, 1100–1102

- security groups, 818
- Security Identifier (SID), 831
- security inheritance, 69
- securityadmin privilege, 158
- self-service site creation
 - administrative tasks and, 271
 - configuring, 283, 329
 - enabling, 797–799
 - functionality, 482
 - permission considerations, 792
- Semantic Model Definition Language (SMDL), 1000
- SendEmail activity, 1095
- Sequential Workflow project template, 1093, 1095
- Server farm account, 149
- server roles. *See also* specific server roles
 - Active Directory and, 141
 - deployment and, 80–82
 - installation requirements, 139–141
- service accounts, 328, 783, 1025–1026
- service applications. *See also* specific service applications
 - configuring, 364–372
 - configuring associations, 286
 - configuring properties, 286
 - configuring SharePoint, 181–182
 - connecting to, 284
 - creating, 284, 581
 - creating instances, 367–369
 - default, 60
 - defined, 60, 283, 347
 - deleting, 284
 - deploying, 365
 - disaster recovery planning, 834, 837
 - identifying administrators, 285
 - identifying service locations, 356–364
 - installing, 179
 - logical architecture, 350–351
 - managing, 285
 - modifying application pools, 369–371
 - permission considerations, 823
 - planning architecture, 357–361
 - provisioning, 179
 - proxy groups, 352–353, 366–367, 371–372
 - publishing, 353, 373–386
 - scaling architecture, 354–355
 - services commonly associated, 347
 - timer jobs, 304–306
 - Web applications and, 351, 365
- service architecture
 - default applications, 60
 - overview, 59–60
- service principal name (SPN), 259
- service proxy/connector, 60
- service token issuer, 381
- services. *See also* specific services
 - activating, 960
 - defined, 347
 - starting/stopping, 364–365
- Services Federation, 347–350
- Set-AuthenticodeSignature cmdlet, 202
- Set-ExecutionPolicy cmdlet, 202, 605
- Set-FASTProp function, 611
- Set-ManagedPropFromName function, 599–600
- Set-SPEnterpriseSearchAdministrationComponent cmdlet, 582
- Set-SPEnterpriseSearchCrawlContentSource cmdlet, 587, 591
- Set-SPEnterpriseSearchCrawlTopology cmdlet, 586
- Set-SPEnterpriseSearchExtendedConnectorProperty cmdlet, 610
- Set-SPEnterpriseSearchIndexPartition cmdlet, 586
- Set-SPEnterpriseSearchQueryAuthority cmdlet, 595
- Set-SPEnterpriseSearchQueryTopology cmdlet, 586
- Set-SPEnterpriseSearchServiceApplication cmdlet, 613
- Set-SPSearchService cmdlet, 574
- Set-SPSearchServiceInstance cmdlet, 574
- Set-SPWeb cmdlet, 239
- setup log files, 1004
- Setup user account, 148
- Setup.exe command, 153–154, 177, 188
- Set-WsManQuickConfig cmdlet, 203
- Shared Services Provider (SSP), 59, 346, 1074–1075
- SharePoint 2010
 - cmdlets for, 231–232
 - comparing search features, 526–528
 - editions supported, 6–8, 136–139
 - enhanced features, 5–6, 8
 - installation considerations, 137, 1054
 - managing services, 237–238
 - middle management and, 79–80
 - platform overview, 4–5
- SharePoint 2010 Management Shell
 - administrative tasks, 232–241
 - cmdlet support, 214–221
 - cmdlets for SharePoint 2010, 231–232
 - cmdlets for SharePoint Foundation 2010, 230
 - functionality, 210–211
 - getting help, 222–228
- SharePoint 2010 Online, 138
- SharePoint administrators, 400

- SharePoint Designer 2010
 - functionality, 7
 - general settings, 268, 337
 - modifying list input forms, 46–49, 74
 - publishing site settings, 740
 - workflow support, 1085–1086, 1089–1090, 1097–1098, 1120
- SharePoint Directory Service, 293
- SharePoint Foundation 2010
 - cmdlets for, 230
 - comparing search features, 526–528
 - Complete installation option, 176
 - enterprise search and, 578
 - functionality, 7–8, 137
 - Standalone installation option, 164
- SharePoint Foundation Subscription Setting Service, 349
- SharePoint Health Analyzer bar, 249
- SharePoint List Filter Web Part, 937
- SharePoint Mobile. *See also* Office Web Applications
 - Microsoft Office Mobile 2010, 954
 - mobile search, 966
 - mobile views, 967
 - overview, 954–955
- SharePoint Workspace Mobile, 955
- SharePoint Portal Server, 1073–1074
- SharePoint Products Configuration Wizard
 - in-place upgrades, 1066
 - Office Web Application support, 959
 - opening page, 158
 - reconfiguring server farms, 189
 - Welcome page, 159
- SharePoint Root, 831
- SharePoint Search Server
 - customizing search pages, 635–655
 - functionality, 616–618
 - improving collection quality, 669–671
 - improving search quality, 656–668
 - improving user experience, 619–635
- SharePoint Workspace, 49–53
- SharePoint Workspace Mobile, 955
- Sharing Center Web Part, 967
- sharing knowledge
 - Outlook Social Connector and, 496
 - overview, 491
 - social features, 491–496
- Short Message Service (SMS), 268, 292
- SID (Security Identifier), 831
- Silverlight Web Part, 14, 749, 950
- Simple Mail Transfer Protocol (SMTP), 254, 267, 292
- Simple Object Access Protocol (SOAP), 289, 945
- simple recovery model, 129
- Site Actions menu
 - depicted, 11, 247
 - Edit Page command, 13
 - functionality, 248
 - New Site command, 12
- Site Aggregator Web Part, 940
- site collection administrators, 282, 801, 1100
- site collections
 - activating features, 961–962
 - application management, 276–283
 - auditing activities, 800–801
 - backup and restore, 322, 863
 - configuring content type publishing, 706–707
 - configuring object cache, 765–766
 - configuring output cache, 762–763
 - configuring quotas and locks, 281–282
 - confirming site use/deletion, 279
 - content deployment and, 771
 - creating, 271, 277
 - creating with multiple databases, 288
 - custom policies, 800
 - defined, 10, 216
 - deleting, 278–279
 - In Place Records Management feature, 429
 - information organization tools, 471
 - managing user solutions, 297–298
 - multi-tenancy, 72
 - Office Web Applications support, 961–962
 - provisioning, 71
 - security considerations, 800–801
 - self-service site creation, 271, 283, 329
 - site columns and, 39–40
 - site model and, 73
 - team collaboration features, 478
 - templated categories, 255
 - throttling thresholds, 36–37
 - viewing, 283
 - workflow usage summary, 1117
- site columns
 - creating, 40
 - defined, 39
 - linking to, 39
 - modifying, 40
- site directory, 336–337, 471
- site model, 73
- site quotas
 - configuring, 281–282
 - default templates, 265
 - defined, 10, 780

- specifying templates, 280
- Site Settings menu, 12
- site templates. *See also* specific templates
 - categorizing, 8–10
 - meeting workspace, 476–477
 - Search Center, 631
 - site collection categories, 255
 - site model and, 73
- Site Users Web Part, 749
- sites
 - adding to portal trunks, 984–988
 - applying UI upgrades, 18
 - blank, 478
 - collaboration capabilities, 76
 - configuring output cache, 763
 - creating, 10–12, 271
 - enabling features, 332
 - exporting, 323, 864
 - identifying workflow roles, 1100
 - importing, 864–865
 - indexing content, 802
 - information organization tools, 471
 - interacting with, 10
 - monitoring workflow status, 1115–1116
 - provisioning, 71
 - security considerations, 10, 801–818
 - site model and, 73
 - types supported, 8–10
 - user permissions, 272
 - workflows by association, 1081, 1112–1113
- Sites In Categories Web Part, 940
- Sites In Category Web Part, 933
- small farm deployment, 83–84, 387, 389–391
- smart phones, 970
- SMDL (Semantic Model Definition Language), 1000
- SMS (Short Message Service), 268, 292, 970–973
- SMTP (Simple Mail Transfer Protocol), 254, 267, 292
- SOAP (Simple Object Access Protocol), 289, 945
- Social Collaboration Web Parts, 943–944
- Social Meeting Workspace template, 477
- social networking capability
 - communities as backbone, 77, 491
 - sharing knowledge via, 491–496
 - Social Tagging database, 103, 501
 - Social Tags And Notes feature, 499
- social search, 77, 495
- social tagging
 - defined, 493
 - Social Tagging database, 103, 501
 - Social Tags And Notes feature, 499
 - user profile support, 498
 - Web Part support, 943
- software requirements
 - application servers, 144
 - client computers, 145–146
 - database servers, 143
 - disaster recovery planning, 837
 - for SharePoint, 141
 - for SQL Server, 111
 - Microsoft Office, 146
 - Microsoft Silverlight, 146
 - operating system, 142–143
 - Web servers, 144
- sorting list data, 33
- SOX (Sarbanes-Oxley Act), 398
- sp_configure stored procedure, 105–106, 115
- spell checking feature, 658–659
- spell-tuning feature, 657–658
- SPFarmConfig cmdlet, 215
- SPN (service principal name), 259
- Sprestore.log file, 852
- SPSite cmdlet, 216
- SPTimerJob cmdlet, 576, 861
- SQL (Structured Query Language), 70
- SQL Server (Microsoft)
 - backup and restore, 870
 - BACKUP LOG command, 128
 - database services and, 62
 - database snapshots, 875
 - default, 289
 - disaster recovery and, 831, 835, 845
 - failover clustering, 875
 - index considerations, 32
 - installing, 107–111, 158
 - optimizing for SharePoint, 112–130
 - storage considerations, 65, 96
 - system databases, 100–102
 - system stored procedures, 105–106
 - transaction log shipping, 872–874
 - Transact-SQL statements, 105–106, 845
 - user databases, 102–104
 - verifying permissions, 1056–1057
 - versions supported, 96–97
 - Windows PowerShell support, 106–107
- SQL Server Analysis Services Filter Web Part, 937
- SQL Server instances
 - default, 98
 - defined, 97
 - multiple, 114–115
 - named, 99

- primary components, 98
- SQL Server Management Studio, 104–105
- SQL Server Reporting Services
 - architectural overview, 998–1007
 - architecture and tool changes, 997
 - clients tier, 1001
 - connecting to, 104
 - data tier, 1005
 - deploying, 1016–1017
 - features supported, 994
 - formats supported, 998
 - installing/configuring, 1007–1016
 - log files, 1004
 - managing reports, 1016–1017
 - memory management, 1003
 - report authoring, 995–996
 - report processing/rendering, 997
 - report server tier, 1001–1004
 - SharePoint tier, 1001
- SQL Server services account, 148
- Sqlcmd.exe, 105
- SQLIO.exe, 132–133
- sqlps utility, 107
- SQLServerAgent service, 98
- SSP (Shared Services Provider), 59, 346, 1074–1075
- SSSC. *See* self-service site creation
- StackSelector Web Part, 948
- Staging database, 103
- Start-Job cmdlet, 208
- Start-ServiceWinRM cmdlet, 203
- Start-SPAssignment cmdlet, 228–229
- Start-Transcripts cmdlet, 233
- State database, 103
- State Machine workflow project template, 1094–1095
- State Service service application
 - cmdlet equivalent, 179
 - databases supported, 103
 - error message, 45–46
 - functionality, 182, 348
- Status Indicator Web Part, 932
- Stop-SPAssignment cmdlet, 228–229
- Stop-Transcripts cmdlet, 233
- stop-words, 619
- storage
 - as core service, 65–68
 - assemblies, 63
 - caching considerations, 754
 - content types, 417
 - database file options, 116
 - disk space options, 130–131
 - for backups, 828

- My Site pages, 492
- SQL Server support, 65, 96
- user profiles, 497
- Storage Area Network (SAN), 116, 832
- Store database, 103
- stored procedures, 105–106
- Structured Query Language (SQL), 70
- STS (Secure Token Service), 891
- STSADM command
 - activating developer dashboard, 312
 - associating databases, 187
 - automating installation, 234
 - backup and restore support, 861–866
 - disaster recovery and, 844–845
 - performing upgrades, 1058–1059
 - security considerations, 822
- style sheets, 17, 720
- subsites
 - exporting, 864
 - importing, 864–865
 - Web pages versus, 736–737
- sub-types, creating, 508–509
- Summary Links Web Part, 749, 940
- Sun Java System Directory Server, 918
- Synchronization database, 103, 501
- synchronizing profiles, 498, 515–519
- syndication, content types, 703–707
- system workflows, 1078

T

- Table Of Contents Web Part, 749, 933, 940
- Tag Cloud Web Part, 943
- Task List View Web Part, 936
- Task Web Part, 14
- taxonomy
 - business, 454
 - defined, 454
 - managing social tagging, 494
 - post-upgrade configurations, 1074
 - predefined vocabularies, 461
 - putability and, 459–461
- Team Collaboration features, 477–478
- Team Site template, 10, 25, 476, 928
- Tempdb system database, 102, 126–127, 131
- term sets, 698–703
- term store, 698–703
- Term Store database, 103
- test environment, 87–88
- testing farm backups, 1047

Test-SPContentDatabase cmdlet, 1057–1058
 Text Filter Web Part, 937
 text messages, setting up SMS alerts, 970–973
 themes
 creating via user interface, 15
 creating with PowerPoint, 15–16
 creating with style sheets, 17
 WCM support, 721, 733
 This Week In Pictures Web Part, 939
 Three-state workflow, 1085
 throttling
 bit rate, 756
 defined, 36
 external system, 910–912
 large list support, 36–37
 managing thresholds, 266–267
 requesting overrides, 36
 setting limits, 38
 time zones (general setting), 265
 timer jobs, 304–306, 1079
 tokenization, 657
 Top Federated Results Web Part, 943
 topology service, 81–87, 382
 trace log files, 1004
 Trace-Command cmdlet, 215
 transaction logs
 backing up, 128
 functionality, 127
 NTFS allocation unit size, 131
 RAID arrays and, 117
 shipping, 872–874
 simple recovery model and, 129
 SQL Server support, 116
 Transact-SQL statements, 105–106, 845
 Translation Management workflow, 1085
 troubleshooting
 backup jobs, 852
 content search issues, 670–671
 crawls, 554–561
 workflow issues, 1119–1120
 trust certificates
 copying, 379
 exporting, 376–378
 importing, 379–381
 trust management, 328, 379–380

U

UAG (Unified Access Gateway)
 alternate access mappings, 979–981

 application portal trunks, 981–988
 authentication support, 974
 cross-firewall access, 979
 functionality, 977–978
 public host names, 979
 setting up, 978–988
 UI. *See* user interface (UI)
 ULSViewer tool, 924
 unattended service accounts, 1025–1026
 Unicode format, 530
 Unified Access Gateway. *See* UAG (Unified Access Gateway)
 Unified Logging Service (ULS), 235, 307
 uninstalling SharePoint, 188–191
 Update-SPInfoPathAdminFileURL cmdlet, 1075
 upgrade and migration. *See also* database attach
 upgrades; in-place upgrades
 applying UI, 18
 best practices, 1044–1045
 checking status, 333
 functionality, 1037–1038
 hybrid upgrade approach, 1072
 managing, 331–333
 PerformancePoint Services, 1029–1033
 philosophy, 1036–1037
 planning, 1036
 post-upgrade configurations, 1074–1075
 preparing for, 1044–1051
 pre-upgrade checker tool, 1047–1051
 primary scenarios, 1038–1040
 special cases, 1040–1042
 steps to successful, 1052
 supported topologies, 1042–1043
 types supported, 1038
 upgrade considerations, 1051–1052
 upgrade paths, 1042
 upgrade restrictions, 1043–1044
 Windows PowerShell support, 238–240
 uploading files, 266, 271
 Usage and Health Data Collection service application
 configuring, 308–310
 databases supported, 102
 functionality, 182, 348
 User Code Service, 298, 357
 user context, 621–625, 630
 user interface (UI)
 applying upgrades to sites, 18
 creating themes via, 15
 digital asset management, 723
 upgrading, 1051

User Profile service application

- User Profile service application
 - architectural overview, 499–501
 - as modular component, 58
 - cmdlet equivalent, 179
 - configuring, 501–515
 - creating information repository, 497
 - creating user profile properties, 509–511
 - creating user sub-types, 508–509
 - creating/configuring audiences, 513–515
 - databases supported, 103, 501
 - delegating administration, 507
 - deleting/editing user profile properties, 512
 - editing, 506
 - editing user profile policies, 512–513
 - functionality, 60, 182, 349, 473
 - post-upgrade configurations, 1074
 - profile synchronization, 515–519
 - service architecture and, 60
 - setting up, 501–515
 - setting up My Site, 519–521
 - starting, 502–506
 - uses and benefits, 497–499
- User Profile Synchronization Service
 - creating connection, 516–517
 - editing connection filters, 518
 - starting, 516
- user profiles
 - adding, 918
 - context information, 620
 - creating properties, 509–511
 - creating user sub-types, 508–509
 - deleting/editing properties, 512
 - editing policies, 512–513
 - external data and, 918–919
 - Profile database, 103, 501
 - social tags, 498
 - storing, 497
 - synchronizing, 498, 515–519
- user sub-types, creating, 508–509
- User Tasks Web Part, 749

V

- versioning
 - as storage service, 66–67
 - defined, 66
 - disaster recovery and, 838–839
 - document management, 678–680
 - settings, 821–822
 - upgrade considerations, 1052

- video streaming, 722
- View Application Pages permission, 810
- View Lists permission, 809
- View Pages permission, 809
- views, 64, 967–969. *See also* list views
- Virtual Private Network (VPN), 975
- Visio (Microsoft), 1085–1088, 1091–1092
- Visio Graphics Service, 60, 182, 349
- Visio Process Repository template, 476
- Visual Studio 2010
 - creating workflows, 1095–1097
 - custom workflows, 1120
 - functionality, 1086
 - modeling tools comparison, 1097–1098
 - workflow activities, 1094–1095
 - workflow project types, 1093–1094
- VPN (Virtual Private Network), 975

W

- WCF (Windows Communication Foundation), 885
- WCM (Web content management). *See also* portal sites
 - accessibility and markup, 720
 - administering publishing sites, 735–744
 - authoring support, 745–752
 - configuring content caching, 752–767
 - Content Query Web Part, 720
 - enhanced features, 720, 734
 - folder support, 721
 - managing content deployment, 767–777
 - overview, 725–726
 - page authoring, 721
 - publishing site types/features, 726–734
 - publishing Web content, 745–752
 - rating content quality, 720
 - themes and branding, 721, 733
- Web Analytics service application
 - configuring, 742–743
 - databases supported, 103
 - functionality, 60, 72, 182, 349
 - reporting capability, 743–744
 - viewing reports, 314
- Web Analytics Web Part, 749, 948
- Web Application Ribbon
 - Anonymous Policy option, 275
 - Extend option, 263
 - functionality, 256
 - General Settings option, 264–269
 - Permission Policy option, 275
 - Service Connections option, 269

- User Policy option, 274
- Web applications. *See also* application management;
- Office Web Applications
 - accessing, 254
 - Alternate Access Mappings, 276, 295
 - application pools and, 255
 - associating with content databases, 1057–1059
 - configuring mobile accounts, 972
 - configuring object cache, 765
 - configuring output cache, 761–762
 - creating, 256–261
 - creating proxy groups, 366–367
 - deleting or removing, 264
 - e-mail settings, 267
 - extending, 263
 - external service connections, 333–335
 - general settings, 264–266, 333
 - information organization tools, 471
 - managed paths, 269
 - managing, 254–255, 268
 - Microsoft Office support, 349
 - mobile accounts and, 268
 - modifying service connections, 269
 - planning for collaboration, 479
 - policies for, 274–276
 - preparing for upgrades, 1054–1055
 - Records Center template, 417
 - resource throttling settings, 266–267
 - search capability, 337–338
 - security considerations, 786–799
 - service applications and, 351, 365
 - SharePoint Designer settings, 268, 337
 - site directory, 336–337
 - specifying user policy, 326
 - workflow settings, 267, 1105–1106
- Web browsers
 - file handling settings, 265
 - online status of users, 265
 - software requirements, 145–146
- Web content management. *See* WCM (Web content management)
- web database sites, 9–10
- Web pages
 - configuring output cache, 763–764
 - mobile view support, 967–969
 - subsites versus, 736–737
- Web Parts. *See also* specific Web Parts
 - adding, 14, 923
 - Advanced category, 926–927
 - Appearance category, 925
 - closing, 923
 - connecting, 927
 - deleting, 923
 - disaster recovery planning, 832
 - editing, 923–927
 - functionality, 63, 921–922
 - in publishing pages, 748–750
 - Layout category, 926
 - managing, 922–927
 - managing security, 329, 796–797
 - Microsoft Office support, 949–950
 - mobile view support, 967–968
 - modifying for forms, 41
 - moving, 923
 - newly released, 14
 - out-of-the-box, 927–944
 - page model and, 740
 - PerformancePoint support, 947–948
 - personalization support, 76
 - removing broken, 924
 - security considerations, 273–274
 - SharePoint Designer support, 46
 - user permissions, 272
 - Web Analytics support, 948
- Web server role
 - disaster recovery planning, 832–833, 836
 - FAST Search Server support, 568
 - functionality, 81, 140, 388
 - hardware requirements, 142
 - identifying service locations, 357
 - mobile devices and, 973
 - planning topology, 361
 - security considerations, 783
 - software requirements, 144
 - time-based execution, 1082
 - User Profile Service and, 500
- WebSSO, 70
- Welcome To Your Site! message, 14
- WF (Workflow Foundation), 62, 254
- Where-Object cmdlet, 226, 236
- WhileActivity activity, 1095
- WIF (Windows Identity Foundation), 258
- wikis, 490, 493
- wildcards
 - asterisk as, 219
 - managed paths, 269
 - SQL Server Reporting Services and, 1002
- Windows Application Log, 1004
- Windows Communication Foundation (WCF), 885
- Windows Identity Foundation (WIF), 258

Windows Performance Logs

- Windows Performance Logs, 1004
- Windows Phone, 969
- Windows PowerShell. *See also* cmdlets
 - activating site collection features, 961
 - administrative tasks, 232–241
 - architectural overview, 197–198
 - backup and restore, 851–860, 866
 - cmdlet case sensitivity, 107
 - command-line shortcuts, 199–200
 - configuring mobile accounts, 973
 - creating BCS applications, 884
 - deploying with FAST Search, 601–613
 - deploying with scripts, 234–237
 - disaster recovery and, 843
 - enterprise search and, 577
 - execution policy, 201–202
 - functionality, 196
 - installing, 200–202
 - large list support, 36
 - limiting commands, 207
 - management services support, 72
 - managing systems remotely, 203–209
 - memory considerations, 228–230
 - new features, 196–197
 - operators, 227
 - performance considerations, 219
 - security considerations, 822
 - service application equivalents, 179
 - setting throttle limits, 38
 - SQL Server support, 106–107
 - upgrade support, 238–240
- Windows PowerShell scripts, 593–594
- Windows SharePoint Services, 289
- Windows Workflow Foundation (WF), 62, 254
- WinRM
 - configuring, 203–205
 - installing, 200–201
 - managing systems remotely, 203–209
- Word (Microsoft), 53, 67
- Word Automation Services service application, 60, 182, 349
- word breakers (search capability), 529
- Word Viewing service application, 963
- Workflow Foundation (WF), 62, 254
- workflows
 - activating, 1108
 - adding associations, 1109–1113
 - alerts and notifications, 731, 1103, 1106
 - approval, 76
 - classifying, 1080–1081
 - compiled, 1080
 - compliance capabilities, 1083
 - creating, 1095–1097
 - declarative, 1080, 1082, 1101–1102, 1108
 - deploying, 1107–1114
 - development tools, 1085–1098
 - functionality, 62
 - human, 78, 1078
 - identifying roles, 1099–1100
 - information disclosure, 1102–1104
 - managing settings, 267
 - modeling tools, 1081, 1085–1098
 - monitoring, 1114–1116
 - performance analysis reports, 1118–1119
 - permissions to start, 1102
 - planning for deployment, 1098–1104
 - planning service application architecture, 358
 - platform improvements, 1081–1083
 - Pluggable Workflow Service, 1082
 - predefined, 1083–1085, 1120
 - publishing, 729–731
 - security considerations, 1100–1102
 - setting up configurations, 267, 1104–1107
 - stages and forms, 1079–1080
 - starting/verifying, 1113
 - system, 1078
 - tasks/history, 1103–1104, 1106–1107
 - time-based execution, 1082
 - troubleshooting issues, 1119–1120
- WSRP Viewer Web Part, 950
- Wurman, Richard Saul, 451

X

- XML (Extensible Markup Language), 289, 887, 945
- XML Viewer Web Part, 749
- XMP (Extensible Metadata Platform), 461
- XOML (Extensible Object Markup Language), 1080
- XRANK method, 667–668

Z

- zones, 274, 787–795

About the Authors

An author and educator specializing in SharePoint Products and Technologies, **Bill English** (MVP) is an industry leader on this exciting product from Microsoft. Since 2000, Bill has authored 13 books on Exchange and SharePoint products. Bill is the co-owner of Mindsharp and the Best Practices Conference. He has presented at more than 20 different conferences since 2001 and more than 20 different user groups in the last three years on the subjects of Search and Findability. You can find his blog at sharepoint.mindsharpblogs.com or follow him on Twitter @MinnesotaBill. Bill lives with his wife and two children in Minnesota where summer is the six best days of the year.

Brian Alderman has a master's degree in Computer Information Systems and is currently employed by Mindsharp, which is located in the Minneapolis–St. Paul area of Minnesota. Since 1995, he has held several Microsoft certifications that currently include MCT, MCTS for SharePoint, MCITP for SQL Server, MCSA and MCSE for Windows Server, and also PMP certification. As a Senior Technical Instructor and Consultant, he has nearly 15 years of experience in networking, SQL Server databases, SharePoint technologies, and project management. Brian's previous publications include *Windows 2000 Professional* and *The SQL Server 2000 Book*. He is an active speaker at SharePoint industry conferences including SharePoint Saturday and the SharePoint Best Practices Conference.

Mark Ferraz is the President of SolutionsMark (www.solutionsmark.com), a Houston-based consulting firm specializing in the design, development, and implementation of enterprise platforms, solutions, and services based on Microsoft SharePoint Products and Technologies. As a Senior Information Architect, Enterprise Network/Systems Architect, and Developer, Mark has more than 10 years of experience designing, managing, and implementing complex projects for some of the world's largest companies. In addition to being a contributing author of *Microsoft Office SharePoint Server 2007 Best Practices*, Mark is an active speaker at industry conferences such as SharePoint Best Practices Conference and SharePoint Technology Conference.

What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

Microsoft
Press

Stay in touch!

To subscribe to the *Microsoft Press*® *Book Connection Newsletter*—for news on upcoming books, events, and special offers—please visit:

microsoft.com/learning/books/newsletter