

Auf einen Blick

1	SharePoint als Entwicklungsplattform	13
2	SharePoint-Entwicklungswerkzeuge	69
3	Das erste SharePoint-Projekt	119
4	SharePoint-Objektmodelle	157
5	Benutzeroberflächen umsetzen	269
6	Programmierkonzepte	379
7	Deployment-Konzepte	429
8	Workflows	461
9	Business Connectivity Services	497

Inhalt

Danksagung	11
------------------	----

1 SharePoint als Entwicklungsplattform	13
---	-----------

1.1	Die SharePoint-Features im Überblick	15
1.2	Welche Produkte gibt es?	17
1.3	Welche Produktversion ist die für Sie richtige?	20
1.4	Was ist neu an SharePoint 2010?	21
1.4.1	Language Integrated Query (LINQ)	21
1.4.2	Verbesserungen an der Datenplattform	22
1.4.3	Business Connectivity Services (BCS)	22
1.4.4	Client-Objektmodell	22
1.4.5	Silverlight-Integration	23
1.4.6	Sandboxed Solutions	23
1.4.7	Entwicklerdashboard	24
1.4.8	Verbesserungen in der Benutzeroberfläche	24
1.4.9	Visual Studio 2010 SharePoint Tools	24
1.5	Einrichtung der Entwicklungsumgebung	25
1.5.1	SharePoint auf einem Windows Server 2008 (R2) installieren	27
1.5.2	SharePoint auf Windows 7 oder Windows Vista SP 1 installieren	31
1.5.3	Visual Studio 2010 installieren	35
1.5.4	SharePoint Designer 2010 installieren	36
1.5.5	Umgebungsvariablen registrieren	37
1.6	Die SharePoint-Serverarchitektur	37
1.7	SharePoint als Entwicklungsplattform	39
1.7.1	Herausforderungen für den SharePoint-Entwickler	39
1.7.2	Den richtigen Einstieg finden	41
1.7.3	SharePoint und ASP.NET 3.5	43
1.7.4	Die SharePoint-Programmierschnittstellen	44
1.7.5	Das SharePoint-Systemverzeichnis	47
1.7.6	SharePoint und die Internet Information Services.....	49
1.8	Fehlersuche und Überwachung	57
1.8.1	SharePoint United Logging Service (ULS)	58
1.8.2	Internet Information Services Logs	60
1.8.3	Windows-Ereignisdatenbank	60

1.8.4	Überwachung von Client-Komponenten	61
1.8.5	Das SharePoint-Entwicklerdashboard	65
1.9	Zusammenfassung	67

2 SharePoint-Entwicklungswerkzeuge 69

2.1	Visual Studio 2010	70
2.1.1	Welche Produktversionen gibt es?	71
2.1.2	SharePoint 2010-Projektvorlagen	71
2.1.3	SharePoint-Elementvorlagen	76
2.1.4	Feature und Package Designer	76
2.1.5	Bereitstellungskonfiguration	80
2.1.6	Solution-Paket importieren	82
2.1.7	SharePoint Server-Explorer	83
2.1.8	Erweiterte Integrationsfunktionen	84
2.2	SharePoint Designer 2010	88
2.2.1	Abgrenzung zu Visual Studio 2010	90
2.2.2	SharePoint Designer aus Sicht des Administrators	91
2.2.3	SharePoint Designer aus Sicht des Anwenders (Power Users)	92
2.2.4	SharePoint Designer aus Sicht des Webdesigners	101
2.3	PowerShell	102
2.3.1	Einführung in PowerShell	103
2.3.2	Cmdlets	105
2.3.3	Das Einmaleins des PowerShell-Scripting	105
2.3.4	PowerShell und SharePoint	110
2.4	Zusätzliche SharePoint-Werkzeuge	112
2.4.1	SharePoint-Standardwerkzeuge	112
2.4.2	Community Tools	113
2.5	Zusammenfassung	117

3 Das erste SharePoint-Projekt 119

3.1	Vorbereitung der Umgebung	119
3.1.1	Leere Websitesammlung erstellen	120
3.1.2	Listen erzeugen	120
3.2	Leeres SharePoint-Projekt erzeugen	121
3.3	Ressourcen hinzufügen	126
3.4	Feature erstellen	127
3.5	Feature Receiver erstellen	129
3.6	Projekt bereitstellen	132

3.7	Visuellen Webpart programmieren	139
3.8	Einen visuellen Webpart debuggen	151
3.9	Fazit	156

4 SharePoint Objektmodelle 157

4.1	Die Auswahl des richtigen Objektmodells?	158
4.2	Server-Objektmodell	159
4.2.1	Assemblies und Namensräume	159
4.2.2	Server-Architektur	160
4.2.3	Site-Architektur	165
4.2.4	Websitesammlungen und Webseiten	166
4.2.5	Listen und Dokumentbibliotheken	178
4.2.6	Benutzer, Gruppen und Berechtigungen	202
4.2.7	Heraufstufung und Impersonifizierung	210
4.3	Client-Objektmodell	214
4.3.1	Struktur des Client-Objektmodells	215
4.3.2	Die Funktionsweise des Client-Objektmodells.....	216
4.3.3	Eine Client-Anwendung – Schritt für Schritt	220
4.3.4	Die Struktur des Client-Objektmodells	222
4.3.5	Daten abfragen	224
4.3.6	Daten manipulieren	230
4.3.7	Webseiten und Listen erstellen	234
4.3.8	Arbeit mit Benutzern und Gruppen	241
4.3.9	Arbeit mit Dokumenten	244
4.3.10	Umsetzung von ECMAScript-Client-Anwendungen	247
4.4	Webservices	252
4.4.1	Standard-SharePoint-Webservices	252
4.4.2	Standard-Webservices verwenden	254
4.4.3	Benutzerdefinierte WCF-Services bereitstellen	261
4.5	Fazit	268

5 Benutzeroberflächen umsetzen 269

5.1	Verbesserungen in der SharePoint-Benutzeroberfläche	270
5.2	Masterseiten	272
5.2.1	Einführung in die Technologie	272
5.2.2	Masterseitentypen	274
5.2.3	Standard-Masterseiten	275
5.2.4	Masterseiten-Token	277

5.2.5	Unterstützung von dynamischen Token für Anwendungsseiten	278
5.2.6	Feste Anwendungsseiten	279
5.2.7	Geschützte Anwendungsseiten	280
5.2.8	Allgemeiner Aufbau einer SharePoint-Masterseite	280
5.2.9	Benutzerdefinierte Masterseite bereitstellen	283
5.3	Ribbon	291
5.3.1	Standardelemente des Ribbons	291
5.3.2	Standard-Ribbon erweitern	292
5.3.3	Vollständige Tabs und Gruppen erzeugen	296
5.3.4	Standardschaltflächen ersetzen und entfernen	300
5.3.5	Ribbon um Code erweitern	302
5.3.6	Debuggen von Ribbon-Erweiterungen	305
5.3.7	Kontextuelle Webparts entwickeln	306
5.4	Benutzerdefinierte Aktionen	318
5.4.1	Element-Kontextmenüs erweitern	321
5.4.2	Websiteeinstellungen um neue Gruppen erweitern	323
5.4.3	Standardfunktionen entfernen	325
5.4.4	Benutzerdefinierte Web-Controls	325
5.5	Statusliste, Benachrichtigungen und Dialoge	327
5.5.1	Statusleiste anpassen	329
5.5.2	Nachrichtenleiste anpassen	332
5.5.3	SharePoint-Dialoge verwenden	333
5.6	Webparts	336
5.6.1	ASP.NET- oder SharePoint-Webparts?	337
5.6.2	Webpart-Definition	338
5.6.3	Webpart-Zonen	339
5.6.4	Personalisierung	339
5.6.5	Der »SafeControl«-Eintrag	341
5.6.6	Der Lebenszyklus von Webparts	342
5.6.7	Webparts und Code Access Security	345
5.6.8	Einen vollständigen Webpart programmieren	349
5.7	Silverlight	355
5.7.1	Architektur	357
5.7.2	Aufbau einer XAP-Datei	358
5.7.3	Asynchrone Datenabfrage	360
5.7.4	Unterschiede zu WPF	363
5.7.5	XAML	364
5.7.6	Cross-Domain-Datenzugriff	367
5.7.7	SharePoint-Silverlight-Webpart	368

5.7.8	Entwicklungsumgebung	370
5.7.9	Eine vollständige Silverlight-Anwendung programmieren	374
5.8	Fazit	377

6 Programmierkonzepte 379

6.1	CAML	379
6.1.1	Sitedefinitionen	379
6.1.2	Listendefinition	383
6.1.3	Programmierung mit CAML	392
6.2	LINQ to SharePoint	395
6.2.1	Einführung in LINQ	395
6.2.2	LINQ Provider	398
6.2.3	»LINQ to SharePoint«-Provider	400
6.2.4	SPMetal	401
6.2.5	Anwendungsbeispiele für »LINQ to SharePoint«	404
6.2.6	Konfliktmanagement	409
6.2.7	Effiziente und ineffiziente Abfragen	413
6.2.8	Auswertung von LINQ-Abfragen	415
6.2.9	LINQ im Client-Objektmodell	416
6.3	REST	419
6.3.1	Die Architektur von REST	419
6.3.2	Operationen	420
6.3.3	Syntax	421
6.3.4	Optionen	422
6.3.5	REST in SharePoint 2010	423
6.3.6	REST-Clients mit Visual Studio implementieren	425
6.4	Fazit	427

7 Deployment-Konzepte 429

7.1	Features	429
7.1.1	Features installieren und aktivieren	431
7.1.2	»Feature.xml«	433
7.1.3	Element-Manifest	436
7.1.4	Feature Receiver	438
7.1.5	Feature Upgrades	439
7.2	SharePoint-Solutions	443
7.2.1	Solution-Manifest	445
7.2.2	Installation und Bereitstellung	447

7.3	Sandboxed Solutions	449
7.3.1	Architektur	451
7.3.2	Webparts in Sandboxed Solutions	453
7.3.3	Installation einer Sandboxed Solution	454
7.3.4	Validierung	456
7.3.5	Monitoring und Ressourcenüberschreitung	458
7.4	Fazit	460
8	Workflows	461
8.1	Architektur	462
8.1.1	Aktivitäten	464
8.1.2	Workflowaufgaben	464
8.1.3	Correlation Token	465
8.2	Workflow-Typen	466
8.3	Workflowformulare	468
8.4	Vergleich der Entwicklungswerkzeuge	469
8.5	Workflows in SharePoint	470
8.6	SharePoint Designer 2010	473
8.7	Workflows mit Visual Studio 2010	479
9	Business Connectivity Services	497
9.1	Architektur	498
9.1.1	Server-Komponenten	499
9.1.2	Externe Inhaltstypen	500
9.1.3	Authentifizierung gegenüber externen Daten	501
9.1.4	Berechtigungen auf Modelle und Entitäten	502
9.1.5	Metadaten-Modell	503
9.1.6	Primärschlüssel	512
9.2	Erstellung von benutzerdefinierten Modellen	514
9.2.1	SharePoint Designer	514
9.2.2	Visual Studio 2010	520
	Index	539