

Dr. Holger Schwichtenberg

Microsoft .NET 3.0 Crashkurs

Dr. Holger Schwichtenberg

Microsoft .NET 3.0 Crashkurs

The Microsoft Press logo is located in the bottom right corner. It consists of the word "Microsoft" in white and "Press" in blue, both in a sans-serif font, set against a black rectangular background.

Microsoft[®]
Press

Dr. Holger Schwichtenberg: Microsoft .NET 3.0 Crashkurs
Microsoft Press Deutschland, Konrad-Zuse-Str. 1, 85716 Unterschleißheim
Copyright © 2007 by Microsoft Press Deutschland

Das in diesem Buch enthaltene Programmmaterial ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor, Übersetzer und der Verlag übernehmen folglich keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programmmaterials oder Teilen davon entsteht. Die in diesem Buch erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Marken und unterliegen als solche den gesetzlichen Bestimmungen. Der Verlag richtet sich im Wesentlichen nach den Schreibweisen der Hersteller.

Das Werk, einschließlich aller Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
09 08 07

ISBN10: 3-86645-501-1
ISBN13: 978-3-86645-501-6

© Microsoft Press Deutschland
(ein Unternehmensbereich der Microsoft Deutschland GmbH)
Konrad-Zuse-Str. 1, D-85716 Unterschleißheim
Alle Rechte vorbehalten

Korrektur: Astrid Schürmann
Fachlektorat: Klaus Löffelmann, ActiveDevelop, Lippstadt (<http://ActiveDevelop.de>)
Satz: Silja Brands, ActiveDevelop, Lippstadt (<http://ActiveDevelop.de>)
Layout: Gerhard Alfes, mediaService, Siegen (www.media-service.tv)
Umschlaggestaltung: Hommer Design GmbH, Haar (www.HommerDesign.com)
Gesamtherstellung: Kösel, Krugzell (www.KoeselBuch.de)

Inhaltsverzeichnis

Vorwort	XXIII
Über den Autor Dr. Holger Schwichtenberg	XXVI
Leser-Service	XXVII
Themeneinschränkungen	XXVIII
Ihre Entwicklungsumgebung für .NET	XXIX
Hinweise zu den Listings	XXXI
1 Einführung	1
Was ist .NET?	2
Definition	2
Ziele von .NET	2
Technische Merkmale des .NET Framework	3
Kernbausteine	4
Standardisierung bei ECMA und ISO	6
.NET Connected	7
Plattformen	7
Microsoft .NET Framework	8
ECMA SSCLI (Rotor)	8
Novell Mono	8
Geschichte und Versionen	9
.NET 1.x	9
.NET 2.0 (Whidbey)	9
.NET 3.0 (WinFX)	9
Erweiterungen zu .NET	10
.NET 3.5 (Orcas)	11
ECMA SSCLI (Rotor)	12
Mono	12
Produkte	12
Notwendige Produkte	12
Optionale Produkte	13
Details zu einigen Produkten	14
Entscheidung zwischen der deutschen und der englischen Version	16
Installation	18
Betriebssysteme, die bereits .NET enthalten	18
Installation des .NET Framework Redistributable	19
Installation von Visual Studio 2005	19

2	World Wide Wings – Das mehrschichtige Fallbeispiel in diesem Buch	23
	Einleitung.....	24
	Szenario	24
	Datenmodell	27
	Mehrschichtarchitektur.....	28
	Objektmodell.....	29
	Implementierung	30
	Softwarevoraussetzungen für das Fallbeispiel	31
	Installation und Konfiguration	32
	Mögliche Fehler.....	32
	Dokumentation	33
	Aktualisierungen.....	34
	Fragen zu diesem Fallbeispiel.....	35
3	Ihre ersten drei .NET-3.0-Anwendungen	37
	Motivation.....	38
	Erstellung einer Konsolenanwendung in C# 2005 an der Kommandozeile.....	38
	Erstellung einer Webanwendung mit Visual Studio 2005	40
	Erweiterung des Beispiels	42
	Schritte für die Erweiterung	43
	Erstellung einer Windows-Anwendung in Visual Basic mit Visual Studio 2005.....	44
4	Grundkonzepte des .NET Framework 3.0.....	49
	Zwischensprache	50
	Anwendungstypen	52
	Laufzeitumgebung	54
	Programmiersprachen.....	55
	Klassenbibliothek.....	59
	Namensräume	60
	Umfang der Klassenbibliothek	64
	FCL versus BCL.....	65
	Implementierung der FCL.....	65
	Objektorientierung.....	66
	Referenztypen.....	67
	Werttypen (Strukturen)	69
	Typnamen und Namensräume (Namespaces).....	70
	Neuheiten im Typkonzept seit .NET 2.0	72
	Softwarekomponentenkonzept	74
	Der Softwarekomponentenbegriff im .NET Framework.....	74
	Aufbau von Assemblies.....	76
	Signierte Assemblies.....	77
	Befreundete Assemblies (Friend Assemblies)	78
	Speicherorte für Assemblies.....	79

Metadaten	79
Komponentenkonfiguration	81
Assembly-Referenzen	82
Anwendungsdienste (Application Services)	83
Installation von .NET-Anwendungen	84
XCopy-Installation	85
Microsoft Windows Installer (MSI)	85
Kommandozeilenwerkzeuge	86
GUI-Werkzeuge	86
No-Touch-Deployment (NTD)	86
Click-Once-Deployment (COD)	86
Weitere Fähigkeiten der Laufzeitumgebung	87
Speicherbereinigung (Garbage Collector)	87
Prozessabgrenzung durch Application Domains	88
Sicherheit (Code Access Security)	90
Mehrschichtige .NET-Anwendungen	94
Benutzerschnittstellensteuerung	95
Geschäftslogik	96
Ressourcenzugriff	96
Verteilte .NET-Anwendungen	97
ASP.NET-basierte XML-Webservices (ASMX)	98
.NET Remoting	98
Windows Communication Foundation (WCF)	100
Zukunft von .NET Remoting	100
Interoperabilität	101
Interoperabilität zu klassischen C-Bibliotheken	101
Interoperabilität zu COM	101
Interoperabilität zu anderen Komponentenplattformen	102
.NET auf 64-Bit-Systemen	102
Entscheidungsdiagramm	103
Datentypen	105
Leistung	105
Kompatibilität zwischen den .NET-Versionen	106
Weiterbetreiben mit der alten CLR	107
Betreiben mit einer neuen CLR	107
Neukompilierung	109
Visual Studio-Kompatibilität	110
Ermitteln der CLR-Version	110
5 Visual Studio 2005	113
Einleitung	114
Grundfunktionen	114
Neuerungen in der Entwicklungsumgebung	115
Produktvarianten	116

Fensterverwaltung	118
Projektverwaltung.....	121
Projektvorlagen	121
Projektmappen.....	122
Projektelemente	124
Ordnerstruktur.....	124
Projekteigenschaften	125
Verweise (Komponenten- und Projektreferenzen)	126
Speichern und Autowiederherstellung	126
Projektkonvertierung	127
Code-Editoren.....	128
IntelliSense-Funktionen.....	128
Farbdarstellung	131
Änderungsverfolgung.....	131
Zeilennummern und Zeilenumbruch.....	132
Coderegionen	132
Refactoring (Umgestalten)	132
Codeformatierung	134
Kommentare.....	134
Hintergrundkompilierung.....	134
Grafische Editoren (Designer)	135
Designer für Windows Forms, WPF und Webforms.....	135
Designer für nichtvisuelle Komponenten	136
Designer für Workflows.....	136
Designer für Klassendiagramme	136
Arbeit mit Datenquellen	137
Datenverbindungen (Data Connections).....	137
Datenquellen (Data Sources)	138
Kompilierung und Ausführung	141
Übersetzungskonfigurationen	141
Fehlerliste	142
Microsoft Build (MS Build).....	143
Debugger	143
Start des Debugger	143
Funktionen im Haltemodus	144
Bearbeiten und Fortsetzen (Edit & Continue)	144
Steuerung der Debugger-Anzeige	145
Direktfenster (Intermediate Window)	147
Objekttestcenter (Object Test Bench, OTB)	147
Einschränkungen auf 64-Bit-Systemen	148
Weitere Funktionen	148
Aufgabenliste	148
Server-Explorer	149

Objektbrowser und Klassenansicht (Object Browser/Class View).....	150
Codedefinitionsfenster (Code Definition Window).....	152
Visual SourceSafe 2005 (VSS).....	152
Clients	152
Erstellen einer SourceSafe-Datenbank.....	153
Integration in Visual Studio 2005.....	154
Visual Studio 2005 Team System (VSTS).....	156
Team Foundation Server (TFS).....	157
Produktvarianten	157
Modellierung.....	158
Statische Codeanalyse.....	159
Testen.....	161
Leistungsmessung	168
 6 Visual Basic 2005 (VB 8.0).....	171
Einleitung	172
Compiler.....	173
Syntaktische Grundlagen	173
Objektorientierung.....	174
Klassen	174
Generische Klassen	177
Partielle Klassen.....	179
Vererbung.....	180
Ereignisse.....	182
Schnittstellen.....	183
Namensräume.....	183
Deklaration von Namensräumen	183
Import von Namensräumen	184
Verweis auf Wurzelnamensräume mit global.....	184
Datentypen.....	185
Typprüfung	185
Arrays.....	185
Wertelose Wertetypen (Nullable Types)	186
Operatoren	187
Schleifen.....	189
Verzweigungen.....	189
Funktionszeiger (Delegates)	189
Annotationen (.NET-Attribute)	190
Fehlerbehandlung.....	191
Eingebaute Objekte und Funktionen.....	192
Kommentare und XML-Dokumentation.....	193
Zeigerprogrammierung (Unsicherer Code)	193

7	C# 2005 (C# 8.0).....	195
	Einleitung.....	196
	Compiler	196
	Syntaktische Grundlagen.....	197
	Objektorientierung.....	197
	Klassen.....	197
	Generische Klassen.....	200
	Partielle Klassen	200
	Vererbung.....	201
	Ereignisse	202
	Schnittstellen	204
	Namensräume	204
	Deklaration von Namensräumen	204
	Import von Namensräumen	205
	Verweis auf Wurzelnamensräume mit global:	205
	Datentypen.....	205
	Zeichenketten	206
	Typkonvertierung.....	206
	Initialisierung	207
	Wertelose Wertetypen (Nullable Types).....	207
	Arrays	209
	Operatoren.....	209
	Schleifen	212
	Verzweigungen	215
	Funktionszeiger (Delegates)	216
	Annotationen (.NET-Attribute).....	217
	Fehlerbehandlung	217
	Eingebaute Objekte und Funktionen	218
	Kommentare und XML-Dokumentation	218
	Zeigerprogrammierung (Unsicherer Code).....	219
8	Konsolenanwendungen.....	221
	Einführung.....	222
	Erzeugen einer Konsolenanwendung.....	222
	Konsolenanwendungen in Visual Studio 2005	223
	Klasse System.Console	224
	Write() und WriteLine().....	225
	Read() und ReadLine().....	225
	Klassen System.Environment.....	226
	Neuerungen in .NET 2.0	226

9	Softwarekomponenten	229
	Der Softwarekomponentenbegriff im .NET Framework.....	230
	Komponententypen	230
	Erstellung von nichtvisuellen Komponenten	232
	Festlegung von Assembly-Eigenschaften	235
	Erstellung einer signierten Assembly	236
	Erstellung einer gemeinsamen Assembly.....	237
	Befreundete Assemblies.....	238
	Interoperabilität mit COM.....	238
	Zugriff von .NET auf COM	238
	Zugriff von COM auf .NET	239
	Konfiguration von Assemblies.....	240
10	.NET-Klassenbibliothek 2.0	243
	Einleitung	244
	System	244
	System.Object	245
	System.Console.....	247
	System.Type	247
	System.Collections	250
	System.IO	251
	Dateisystem	251
	Dateiinhalte	254
	System.Configuration	255
	Neuheiten ab .NET 2.0	255
	ConfigurationManager.....	256
	Verschlüsselte Sektionen	257
	Anwendungseinstellungen (Application Settings).....	259
	Nutzung der Anwendungseinstellung	260
	System.Diagnostics.....	261
	Prozesse.....	262
	Ereignisprotokolle.....	264
	System.Net.....	266
	HTTP- und FTP-Unterstützung.....	266
	System.Net.NetworkInformation.*	269
	Netzwerkstatus.....	270
	System.Net.Mail und System.Net.Mime	271
	System.Text.....	272
	Textcodierung.....	272
	Reguläre Ausdrücke	272
	System.DirectoryServices.....	273
	Allgemeines Objektmodell.....	274
	Objektmodell für die Suche	275

System.DirectoryServices.ActiveDirectory	277
System.DirectoryServices.Protocol	278
System.Management.....	278
System.Resources	281
Erstellung von Ressourcendateien	282
Zugriff auf Ressourcendateien	282
Streng typisierte Ressourcen	284
System.Security	284
System.Security.SecureString	285
System.Security.AccessControl.*	286
 11 Datenzugriff mit ADO.NET 2.0	291
Einführung.....	292
Neuerungen im Überblick.....	292
Die ADO.NET-Architektur	294
Datenprovider von Microsoft	295
Datenprovider von anderen Herstellern	296
Ermittlung der installierten Datenprovider	296
Der Weg der Daten von der Datenquelle bis zum Verbraucher	297
Datenbankverbindungen (Connection)	299
Verbindungen aufbauen und schließen	299
Verbindungspooling.....	300
Verbindungszeichenfolgen zusammensetzen mit dem SqlConnectionStringBuilder.....	301
Verbindungszeichenfolgen aus der Konfigurationsdatei auslesen.....	302
Ermittlung der verfügbaren Microsoft SQL Server	302
Datenbankbenutzerkennwörter ändern.....	302
Daten lesen mit einem Datareader	303
Ablauf	305
Beispiel.....	305
Behandlung von Null-Werten	306
Multiple Active Results Sets (MARS)	307
Daten lesen und verändern mit einem Dataset	308
Das Objektmodell	309
Daten lesen mit DataSets	310
Datensichten (Dataviews).....	314
Daten ändern mit Datasets	315
Typisierte DataSets (Typed DataSets)	318
Umwandlung zwischen DataSet und XML.....	319
Umwandlung zwischen Dataset und Datareader.....	321
Serialisierung und Remoting für DataSets	321
Befehlsausführung mit Command-Objekten	322
Methoden der Befehlsklassen.....	322
Transaktionen	323

Parameter für Befehle	324
Asynchrone Befehlsausführung	325
Datenproviderunabhängiger Datenzugriff durch Providerfabriken.....	328
Benachrichtigungen über Datenänderungen (Query Notifications)	330
Massenkopieren (Bulkcopy/Bulkimport)	333
Providerstatistiken.....	337
Datenbankschema auslesen	338
Positionierung von ADO.NET und Ausblick	339
 12 Datenzugriff mit System.Xml 2.0	 341
XML-Programmierung	342
Neuerungen in .NET 2.0	342
Neuerungen in .NET 3.0	342
XML-DOM	343
Alternative XML-Zugriffsformen	345
XmlReader.....	346
XmlWriter.....	349
XPathNavigator (XPath Data Model)	351
Ableiten eines Schemas aus XML-Dokumenten.....	353
XML Style Sheet Transformations (XSLT).....	354
 13 .NET Remoting	 357
Einleitung	358
Neuerungen in .NET 2.0	358
Formatters und Channels.....	358
Hosting.....	361
Art der Objektnutzung	361
Channel-Erzeugung	362
Aktivierung und Lebenszyklus.....	363
Remoting-Beispiel	367
Konfiguration.....	368
Quellcode.....	368
Ergebnisse.....	375
Deklarative Remoting-Konfiguration	379
 14 ASP.NET-basierte XML-Webservices (ASMX)	 381
Einleitung	382
Neue Funktionen	382
Webservices erstellen mit ASP.NET 2.0.....	383
Webservices nutzen.....	387
Generierung der Proxy-Klasse	388
Aufruf der Proxy-Klasse.....	389
Gemeinsame Proxy-Typen (Proxy Type Sharing)	390

Asynchroner Aufruf	390
Authentifizierung.....	391
Weitere Möglichkeiten	392
RAD-Datenbindung für von Webservices gelieferte Daten	392
Fehlerbehandlung.....	392
Generierung der Server-Schnittstellen	393
Statische Generierung von Serialisierungs-Assemblies (sgen.exe).....	394
Benutzerdefinierte Serialisierung	395
Zustandsbehaftete Webservices.....	395
Web Services Enhancements (WSE) 3.0	396
Vergleich zwischen .NET Remoting und ASP.NET-basierten XML-Webservices	397
 15 Windows Communication Foundation (WCF)	399
Einleitung.....	400
Architektur.....	400
Bindungen (Binding)	402
Assemblies.....	404
Kompatibilität.....	404
Werkzeuge.....	405
Visual Studio-Integration	405
ServiceModel Registration Tool	405
Service Configuration Editor	405
WCF Service Trace Viewer.....	406
COM+ Service Model Configuration Tool.....	406
WS-AtomicTransaction Configuration Utility	406
Implementierung von WCF-Diensten	406
Dienstklasse	406
Datenklassen.....	408
Erstellung einer Endpunktkonfiguration	415
Bereitstellen eines Metadaten-Dienstes.....	420
Konfiguration der Fehlerübermittlung.....	421
Hosting (WCF-Server-Prozess).....	421
Best Practices: Selbsttest.....	426
Erstellung eines WCF-Clients	426
Vorüberlegungen	426
Erstellen eines Proxy	427
Konfigurationseinstellungen.....	428
Auswahl der Konfiguration.....	429
Aufruf eines WCF-Servers.....	429
Steuerung der Proxy-Klasse	429
Erweitern der generierten Klassen	430
WCF-Sicherheit.....	431
Sicherheitsmechanismen	431
Sicherheitsmodi.....	431

Authentifizierungsverfahren	431
Beispiel	432
Übermittlung der Identität	432
Ermitteln der aktuellen Identität	433
Nutzung der Identität	433
Zugriffsrechte.....	433
Protokollierung.....	434
Beispiele in World Wide Wings	437
Weitere Funktionen.....	440
Funktionen des Service Model Metadata Utility Tool (svcutil.exe)	440
Ein-Weg-Kommunikation (One-Way)	441
Duplex-Kommunikation.....	442
Asynchrone WCF-Aufrufe	443
Sitzungen	444
Gemeinsame Proxy-Typen (Proxy Type Sharing)	444
Weitere Möglichkeiten.....	446
Fazit	447
 16 Windows Workflow Foundation (WF).....	449
Einleitung	450
World Wide Wings-Buchungs-Workflow	450
Architektur	452
Bestandteile des Workflow-Konzeptes	453
Workflow-Formate.....	454
Workflow-Werkzeuge	454
Designer	455
Debugging	456
Kommandozeilenwerkzeuge	457
Workflow-Arten	457
Workflow-Aktivitäten.....	458
Aktivitätsbaum	458
Überblick über die mitgelieferten Aktivitäten	458
Weitere Aktivitäten.....	459
Zustandsmodell	459
Workflow-Hosting	460
Server-Prozess.....	461
WF-Dienste konfigurieren und der Laufzeitumgebung hinzufügen.....	461
Start der Workflow-Laufzeitumgebung.....	461
Workflows erzeugen und starten	462
Ereignisse der Laufzeitumgebung behandeln	462
Steuerung von Workflows	463
Datenaustausch mit Workflows.....	463
Interne Kommunikation über Attribute und Abhängigkeitseigenschaften	463
Parameteraustausch mit dem Workflow-Host	465

Datenaustauschdienste.....	466
Kommunikation mit Webservices	470
Aufruf von Webservices.....	470
WorkflowQueuingService	473
Bedingungen.....	473
Regelsätze (Rule Sets)	474
Persistenz.....	476
Anlegen der Datenbank	477
Aktivierung des Persistenzdienstes in einem WF-Host.....	479
Persistierung eines Workflows	480
Form der Persistierung	480
Laden eines persistierten Workflows.....	481
Ablaufverfolgung (Tracking).....	481
Anlegen der Datenbank	482
Aktivieren des Ablaufverfolgungsdienstes.....	484
Umfang der Ablaufverfolgung.....	484
Auslesen der Ablaufverfolgungsdaten.....	484
Scheduling.....	486
Fehlerbehandlung	486
Transaktionen.....	487
Kompensation	488
Weitere Möglichkeiten	489
 17 Webanwendungen mit ASP.NET 2.0-Webforms	 491
Einleitung.....	493
Überblick über ASP.NET-Webforms.....	493
Neuheiten in ASP.NET 2.0-Webforms	494
Beispielanwendung	496
Das Webform-Programmiermodell	497
Abstraktion durch Serversteuerelemente (Adaptive Rendering)	499
Kompatibilität zu Windows Forms.....	499
Objektorientierung	500
Steuerelement-Hierarchie (Objektbaum)	502
Ereignisbehandlung.....	502
Seitenübergänge	504
Trennung von Layout und Programmcode	505
Übersetzungsvorgang und Verbreitung	506
Steuerelement-Typen	510
Bestandteile eines Webprojekts	512
Entwicklungsumgebung Visual Web Developer.....	513
Grundfunktionen.....	513
Neuerungen im VWD	514
Migration	515

Webprojektmodelle.....	516
Arbeit mit dem Webform-Designer.....	524
Übersetzung von Webanwendungen.....	531
Start von Webanwendungen.....	531
Webserver für ASP.NET.....	532
ASP.NET Development Server.....	533
Internet Information Server (IIS).....	534
Serversteuerelemente.....	534
Einfache Steuerelemente.....	534
Validierungssteuerelemente (Validatoren).....	536
Komplexe Steuerelemente.....	538
Selbst definierte Steuerelemente.....	540
Steuerelemente für kleine Endgeräte.....	545
Steuerelemente von Drittanbietern.....	545
Datensteuerelemente und Datenbindung.....	545
Datensteuerelemente.....	546
Datenbindung.....	547
Datenquellensteuerelemente.....	548
Halbautomatische Datenbindung.....	553
GridView.....	554
DataList.....	557
DetailsView und FormView.....	558
TreeView.....	558
Benutzerdefinierte Datenbindung.....	560
Webparts.....	560
Bestandteile eines Webportals.....	561
Inhalt einer WebPartZone.....	561
Anzeigemodi.....	561
Eigene Webparts erstellen.....	562
Konfiguration.....	563
XML-Konfigurationsdateien.....	563
Administration.....	564
Nutzung von Softwarekomponenten.....	568
Interaktion zwischen ASP.NET und dem Webserver mit den eingebauten Objekten....	568
Nutzung von anderen .NET-Softwarekomponenten.....	569
Zustandsverwaltung.....	569
Zustandsverwaltung auf Sitzungsebene.....	570
Zustandsverwaltung auf Seitenebene.....	571
Zustandsverwaltung auf Anwendungsebene.....	572
Anwendungsbeispiel.....	572
Logik auf dem Client und AJAX.....	573
Einfache Client-Skripte.....	573
Client-Rückrufe (Script Callbacks).....	574

Navigation	574
Navigationselemente	575
URL-Rewriting	576
Sicherheit	578
Authentifizierung	578
Benutzerverwaltung	579
Anwendungsidentität	581
Layout-Vorgaben	585
Vorlagenseiten	585
Layout-Themen (Themes)	586
Individuelle Inhalte	588
Profildaten	588
Mehrsprachige Websites (Lokalisierung)	589
Zwischenspeicherung (Caching)	595
Seiten-Caching	595
Fragment-Caching	595
Cache Invalidation durch die Datenbank	596
Datenquellensteuerelement-Caching	598
Installieren und Konfigurieren von Webanwendungen	599
Weitere Möglichkeiten von ASP.NET	599
Vergleich mit der Java-Welt	600
 18 Windows-Oberflächen mit Windows Forms 2.0	 601
Einleitung	602
Überblick über Windows Forms	602
Funktionsüberblick	602
Windows Forms entwickeln mit Visual Studio 2005	603
Neuheiten in Windows Forms 2.0	604
Funktionen der World Wide Wings-Desktop-Anwendung	606
Anwendungsfälle	606
Gezeigte Funktionen	607
Architektur	607
Hauptmenü	608
Konfiguration	609
Stammdatenverwaltung	610
Call Center-Anwendung	611
Fenster	613
Vorgefertigte Formulare	614
Anzeigen eines Fensters	614
DoEvents	615
Das Visual Basic 2005-Anwendungsmodell	616
Fenster mit Unterfenstern (Multi-Document-Interface)	617
Visuelle Vererbung	619
Dialogfenster	620

Steuerelemente.....	621
Allgemeine Eigenschaften von Steuerelementen.....	621
Steuerelementhierarchie.....	622
Liste der verfügbaren Steuerelemente	622
Container	624
Menüs.....	625
Symbolleisten.....	626
Statusleiste	626
Datenbindung	627
Datenbindung per Rapid Application Development (RAD).....	627
Datenbindung an Objektmengen (Mehrschichtiges Datenbinden)	629
Individuelle Datenbindung mit DataBindings	630
Manuelle Datenbindung	631
DataGridView	631
Zeichnen mit GDI+.....	632
Drucken	634
PrintDocument.....	634
PrintPreviewDialog.....	635
PrintDialog.....	635
Berichterstellung mit Report-Generatoren	636
Mehrsprachige Anwendungen (Lokalisierung)	636
Drag & Drop (»Ziehen & Fallenlassen«).....	638
Zwischenablage.....	639
Weitere Möglichkeiten von Windows Forms	640
Systeminformationen	640
XP Visual Styles	640
Erstellung von Windows-Steuerelementen (Benutzersteuerelemente)	640
Nutzung von Windows Forms-Steuerelementen im Internet Explorer	641
Click-Once-Deployment	642
 19 Windows Presentation Foundation (WPF).....	 645
Überblick über WPF	646
Funktionen der World Wide Wings-WPF-Anwendung	650
Extensible Application Markup Language (XAML).....	651
XAML-Grundkonzepte	651
XAML-Serialisierung.....	653
XAML-Namensräume.....	653
Verbindung von XAML und Programmcode (Code-Behind)	654
Abhängigkeitseigenschaften (Dependency Properties).....	656
XAML Markup Extensions.....	657
XAML-Editoren für WPF	657
WPF-Designer für Visual Studio	657
Microsoft Expression Blend.....	659

XAMLPad	663
Andere Anbieter.....	664
Anwendungen und Fenster	665
WPF-Anwendungsobjekt	665
Fenster	666
Dialogfenster	667
Visuelle Elemente und Steuerelemente (Controls).....	667
Allgemeine Eigenschaften von visuellen Elementen	669
Liste der verfügbaren Steuerelemente.....	669
Panel-Elemente	670
Steuerelementhierarchie	673
Ereignissystem.....	674
Routed Events.....	674
Trigger	675
Befehlssystem	675
WPF-Befehlsbindungen (Command Binding)	676
Vordefinierte Befehle	677
Datenbindung.....	679
Datenbindungsziele	679
Datenquellen.....	679
Datenbindungsrichtung.....	679
Beispiel.....	680
Formatvorlagen (Styles)	682
Definition einer Formatvorlage auf Fensterebene	682
Definition einer Formatvorlage auf Anwendungsebene.....	683
Formatvorlagen und Trigger.....	684
Transformationen	684
Animationen.....	685
Zeichnen.....	688
Weitere Möglichkeiten von WPF.....	688
 20 Enterprise Services und Transaktionen.....	 689
Serviced Components	690
Dienste	690
Voraussetzungen und Optionen	690
Registrierung der Assembly im Application Server	691
COM+-Transaktionsdienste.....	692
Beispiel.....	693
Automatische Feststellung des Endzustands (AutoComplete).....	696
Fernauf Ruf von Serviced Components	696
Services without Components (SWC)	696
System.Transactions	698
Implizite Transaktionen mit TransactionScope.....	698

.NET-Klassen als Teilnehmer von Transaktionen	699
Automatische Umschaltung zwischen lokalen und verteilten Transaktionen (Explizite Transaktionen)	699
21 Zusatzkomponenten	701
.NET Enterprise Library	702
Bestandteile	702
Status	703
Installation	703
Windows PowerShell (WPS)	704
Status	705
Commandlets und Objekt-Pipelining	705
Navigationsprovider	707
Skripte	707
Sicherheit	707
Erweiterungen	707
Weitere Informationen	708
AJAX-Erweiterungen für ASP.NET	708
Web 2.0 und AJAX	708
Microsoft-AJAX Produkte	709
Installationspakete und Quellcode	710
Visual Studio-Integration	711
ScriptManager	711
JavaScript-Erweiterungen	711
Partielle Seitenerzeugung	711
Webservice-Aufrufe	717
Extender	722
Ausblick	723
22 Ausblick und Fazit	725
.NET 3.5 und Visual Studio 9.0 »Orcas« – ein Blick in die Zukunft	726
Neue Funktionen im .NET Framework	726
Neue Funktionen in Visual Studio	727
Bewertung: Wo steht .NET?	728
23 Die Entwicklergemeinde und andere Informationsquellen	731
Zeitschriften	732
Zeitschriften mit dem Schwerpunkt ».NET«	732
Zeitschriften, die regelmäßig über .NET berichten	733
Bücher	733
Newsgroups, Foren und Weblogs	733

Websites.....	734
Angebote von Microsoft	734
Codezone.de	734
Codezone Premier Sites	734
dotnetframework.de	736
Weitere Websites	736
Organisationen.....	737
.NET Code Wise Community.....	737
International .NET Association (INETA).....	737
Microsoft Most Valuable Professionals (MVPs).....	737
Regional Directors	737
Community Leader/Influencer Program (CLIP)	738
Veranstaltungen	738
Schulungen und Workshops.....	738
Konferenzen	738
User Group Treffen	739
Feedback an Microsoft	739
 Literaturverzeichnis	 741
Stichwortverzeichnis	747

Vorwort

Liebe Leserin, lieber Leser,

genau ein Jahr nach .NET 2.0 ist am 6. November 2006 die dritte Hauptversion des .NET Framework erschienen. .NET 3.0 ist eine Erweiterung zu .NET 2.0: Die Laufzeitumgebung, die Sprachsyntax und die Basisklassenbibliotheken von .NET 3.0 sind die gleichen wie in .NET 2.0. .NET 3.0 bietet aber neue Bibliotheken mit noch mehr Klassen an. Dementsprechend ist das vor Ihnen liegende Buch ».NET 3.0 Crashkurs« eine Erweiterung des Vorgängerwerks ».NET 2.0 Crashkurs«.

Besprochene Version

Dieses Buch basiert auf der endgültigen Version des .NET Framework 3.0 vom 6. November 2006 sowie auf Visual Studio 2005 mit den zugehörigen .NET 3.0-Erweiterungen. Bitte beachten Sie, dass zum Redaktionsschluss dieses Buches einige der Werkzeuge für .NET 3.0 erst in einer Alpha-Version vorlagen. Microsoft hat sich entschlossen, die Werkzeuge für die neuen Bibliotheken Windows Presentation Foundation (WPF) und Windows Communication Foundation (WCF) erst etwa ein Jahr später (also Ende 2007) erscheinen zu lassen.

Zielsetzung: Schnelleinstieg

Zielsetzung dieses Buches ist es, die zentralen Konzepte von .NET 3.0 zu beschreiben und anhand ausgewählter Beispiele den Einsatz von .NET 3.0 darzustellen. Sie können von diesem Werk einen kompakten Überblick erwarten, mit dem Sie schnell in das gesamte Themengebiet hineinfinden. Das Buch liefert neben der Beschreibung der Funktionen auch zahlreiche Beispiele und eine Bewertung des aktuellen Zustands der Produkte. Dieses Werk ist unabhängig von den Produktabteilungen und dem Vertrieb im Hause Microsoft entstanden.

Zielgruppe: Ein- und Umsteiger

Der ».NET 3.0 Crashkurs« richtet sich sowohl an Entwickler, die bereits mit .NET 1.x/2.0 entwickelt haben, als auch an jene, die neu in die .NET-Technologie einsteigen möchten. Dementsprechend beschränkt sich dieses Buch nicht auf die Neuerungen, sondern betrachtet das Gebiet ».NET« insgesamt. Um die bereits mit .NET 1.x/2.0 erfahrene Zielgruppe anzusprechen, gilt den Veränderungen und Neuerungen in vielen Beispielen ein besonderes Augenmerk. Zudem sind alle Neuerungen explizit im Text hervorgehoben. Wenn Sie noch nicht mit .NET gearbeitet haben, sollten Sie für die Lektüre dieses Buches gute Vorkenntnisse in C++, Delphi, Visual Basic, Java oder einer anderen objektorientierten Sprache haben. Außerdem sollten Sie die grundsätzliche Funktionsweise einer integrierten Entwicklungsumgebung und visueller Gestaltungswerkzeuge kennen. Dieses Buch eignet sich nicht für Leute, die Programmierung erst von Grund auf erlernen wollen.

Keine Vollständigkeit

Nach der vorherigen Auflage habe ich Kritiken erhalten, die sich darauf bezogen, dass bestimmte Funktionen von .NET in diesem Buch entweder zu kurz oder gar nicht dargestellt wurden. An einigen einfachen Zahlen möchte ich aufzeigen, wie komplex .NET ist: Das .NET Framework 3.0 besteht aus 8714 öffentlichen (d.h. vom Entwickler nutzbaren) Klassen (.NET 2.0: 5684). Hinzu kommen die nicht so leicht zählbaren Funktionen der Werkzeuge (wie Visual Studio) und ergänzende .NET-basierte Bibliotheken (z.B. von Microsoft SharePoint und Microsoft SQL Server). Denjenigen, die sich darüber beschwerten, dass etwas in dem Buch nicht (ausführlich genug) enthalten sei, möchte ich die Gegenfrage stellen: Wie soll es möglich sein, so viele Funktionen auf 794 Seiten auch nur zu erwähnen?

Dieses Buch kann gar nicht den Anspruch auf Vollständigkeit erheben. Das kann kein .NET-Buch. Das vorliegende Buch zeigt einen repräsentativen und praxisnahen Ausschnitt. Ich erwähne in diesem Buch im Wesentlichen die Funktionen, die die meisten Entwickler in den ersten Monaten mit .NET benötigen werden. Dabei zeige ich diese Funktionen nicht vollständig, sondern exemplarisch. Sicherlich ist die Auswahl der Funktionen durch meinen persönlichen Alltag als .NET-Softwareentwickler geprägt.

Im ».NET 3.0 Crashkurs« stehen Konzepte im Vordergrund, nicht kleinste Details. Dieses Buch ist aufgrund seines Umfangs mit Sicherheit nicht unbedingt als Nachschlagewerk geeignet, sondern dient dem Einsteiger als Erläuterung der wichtigsten Konzepte und Werkzeuge.

Komplexeres Fallbeispiel

Viele Fachbücher verwenden kleine, isolierte Beispiele auf »Hello World«-Niveau, die zwar einfach zu überschauen sind, aber weder die Architektur einer komplexen .NET-Anwendung noch das Zusammenspiel verschiedener Techniken aufzeigen. Dieses Buch unterscheidet sich deutlich von diesem »Hello World«-Ansatz, indem in weiten Teilen ein durchgehendes Fallbeispiel verwendet wird. Das Fallbeispiel ist mehrschichtig (mit mehr als drei Schichten) aufgebaut und besitzt mehrere verschiedene Clients (Windows, Web, Befehlszeile). Das Fallbeispiel hat den Vorteil, dass Sie im Laufe des Buches die Daten und Klassen wiedererkennen werden und sich ein großer Grad von Wiederverwendbarkeit in dem entwickelten Code ergibt. Als Nachteil hat sich bei der Voraufgabe gezeigt, dass weniger erfahrene Softwareentwickler Schwierigkeiten hatten, in das Beispiel hineinzufinden. Ich habe mich aber entschlossen, mit diesem Buch auf die Zielgruppe zu fokussieren, die sich ein großes zusammenhängendes Beispiel wünscht. Mit einem Produkt alle Zielgruppen ansprechen zu wollen ist meist kein erfolgreicher Ansatz.

Die Qual der Wahl: Programmiersprache und Werkzeuge

Für einige Softwareentwickler steht die Entscheidung fest, andere hingegen haben die Qual der Wahl zwischen den verschiedenen .NET-Programmiersprachen und Entwicklungswerkzeugen. Auch Autoren von .NET-Fachbüchern stehen vor dieser Wahl. Nur sehr selten lassen es die wirtschaftlichen Rahmenbedingungen zu, ein Buch in mehreren Varianten (mit verschiedenen Sprachen und Werkzeugen) zu erstellen.

Für dieses Buch gab es nur die Möglichkeit, die beiden primären .NET-Sprachen (C# und Visual Basic .NET) in einem einzigen Buch zu mischen und sowohl Visual Studio 2005 als auch das »rohe« .NET Framework 3.0 zu behandeln. Primär befasst sich das Buch jedoch mit C# und Visual Studio, weil die meisten .NET-Entwickler inzwischen diese Kombination einsetzen.

Alle Leser, die lieber mit Visual Basic .NET arbeiten, möchte ich darauf hinweisen, dass die Transformation zwischen Visual Basic und C# sehr einfach ist: Es existieren kostenlose Werkzeuge, die dies auf Mausklick erledigen (siehe <http://www.dotnetframework.de/tools.aspx>). Ich selbst arbeite übrigens im Entwickleralltag zu 75% mit C# und zu 25% mit Visual Basic .NET.

Das .NET Framework und das zugehörige SDK sowie die sogenannten »Express-Editionen« der Entwicklungsumgebung Visual Studio 2005 sind kostenlos auf der Microsoft Website verfügbar. Für alle Leser, die Visual Studio 2005 vor dem Kauf evaluieren möchten, liegt diesem Buch eine DVD mit einer 180-Tage-Lizenz der »allumfassenden« Visual Studio 2005 Team Suite bei.

Deutsche Version

Dieses Buch beschreibt die deutsche Version von Visual Studio 2005, weil der Verlag Microsoft Press davon ausgeht, dass Käufer eines deutschsprachigen Buches auch die deutsche Version von Visual Studio einsetzen. Ich weiß jedoch, dass viele deutsche Entwickler (einschließlich meiner eigenen Person) die englische Version der Software bevorzugen, da die Übersetzungen ins Deutsche oft holprig sind und die Fehlermeldungen nur schwerer verständlich machen. Als Kompromiss zwischen dem Wunsch des Verlags und dem Kundenkreis mit englischsprachiger Entwicklungsumgebung wird dieses Buch – so oft es möglich ist, ohne den Lesefluss zu stören – sowohl die deutschen als auch die englischen Namen der Menüpunkte nennen. Die Bildschirmabbildungen sind aus Platzgründen allerdings immer nur in einer Sprache (i.d.R. Deutsch) abgedruckt.

Zu den .NET 3.0-Erweiterungswerkzeugen für Visual Studio 2005 gab es bis zum Redaktionsschluss noch keine deutschen Versionen bzw. keine Versionen, die in der deutschen Version von Visual Studio überhaupt lauffähig wären. Daher sind diese Teile des Buches komplett mit der englischen Version von Visual Studio beschrieben.

Weiterhin möchte ich noch darauf hinweisen, dass die Anordnung der Menüs und auch einige Tastaturkürzel von den gewählten Einstellungen in Visual Studio 2005 abhängen. Alle Ausführungen in diesem Buch beziehen sich auf die Umgebungseinstellung »Allgemeine Entwicklungseinstellungen«, die bei der Installation des Produkts ausgewählt werden kann.

Dank: Allen, die mitgeholfen haben

Meinen Dank für ihre Mitwirkung an diesem Buch möchte ich aussprechen an

- den Microsoft Press-Lektor *Thomas Braun-Wiesholler*, der dieses Buch ermöglicht hat;
- den Fachlektor *Klaus Löffelmann*, der alle Texte inhaltlich geprüft und das Buch typografisch gesetzt hat;
- die Korrektorin *Astrid Schürmann*, die das Buch sprachlich verbessert hat und
- den Chef des ASP.NET-Entwicklungsteams, *Scott Guthrie*, und an sein Entwicklungsteam, die mir den direkten Zugang zu Software und Informationen ermöglicht haben.

Viel Erfolg beim .NET-Programmieren mit diesem Buch wünscht Ihnen

Dr. Holger Schwichtenberg

Essen, im März 2007

Über den Autor

Dr. Holger Schwichtenberg



- **Ausbildung:**
 - Studium Diplom-Wirtschaftsinformatik an der Universität Essen
 - Promotion an der Universität Essen im Gebiet komponentenbasierter Softwareentwicklung
 - Selbstständig im Bereich Softwareentwicklung seit 1996
 - Aktuelle Tätigkeit:
 - Leitung der Firma www.IT-Visions.de
 - Softwareentwicklung und Softwarearchitektur im Kundenauftrag
 - Beratung und Schulung von Softwareentwicklern
 - Individueller Support
- **Kernkompetenzen:**
 - Objektorientierung, Komponentenorientierung, Serviceorientierung
 - Softwarearchitektur, Mehrschichtige Softwareentwicklung, Verteilte Systeme
 - .NET Framework, ASP/ASP.NET, Visual Studio
 - C#, Visual Basic (VB/VB.NET/VBA/VBS)
 - Component Object Model (COM)
 - Windows Scripting, Microsoft Powershell
 - Active Directory-Programmierung
 - Windows Management Instrumentation (WMI)
 - Relationale Datenbanken, XML
- **Veröffentlichungen und Vorträge:**
 - 21 Fachbücher bei Addison-Wesley, Microsoft Press und dem WEKA-Verlag
 - Mehr als 400 Beiträge in Fachzeitschriften
 - Ständiger Mitarbeiter der Zeitschriften iX, dotnetpro und Windows IT Pro
 - Sprecher auf nationalen und internationalen Fachkonferenzen (z. B. TechEd, OOP, Advanced Developers Conference, Microsoft IT Forum, Wirtschaftsinformatik, Net.Object Days, VS One, Online, Windows Forum, DOTNET-Konferenz, BASTA, XML-in-Action)
- **Ehrenamtliche Community-Tätigkeiten:**
 - Vorstandsmitglied bei codezone.de
 - Sprecher für die International .NET Association (INETA)
 - Betrieb der Community-Websites www.dotnetframework.de und www.windows-scripting.de

- Zertifikate und Auszeichnungen von Microsoft:
 - Most Valuable Professional (MVP)
 - Microsoft Certified Solution Developer (MCSD)
 - .NET Code Wise Community-Experte
 - Codezone Premier Site Member
- Firmen-Website:
 - <http://www.IT-Visions.de>
- Weblog:
 - <http://www.dotnet-doktor.de> (bei Heise.de)
- Kontakt:
 - hs@IT-Visions.de

Leser-Service

Den Lesern dieses Buches werden vom Autor folgende Serviceleistungen im Rahmen einer zugangsbeschränkten Website angeboten:

- **Downloads:** Laden Sie alle in diesem Buch vorgestellten Codebeispiele herunter. Es war leider nicht möglich, die Beispiele mit auf die beiliegende Visual Studio-DVD zu brennen, weil DVD-Produktionsrichtlinien der Microsoft Corporation dies nicht zulassen. Einen zweiten Datenträger konnte Microsoft Press aber nicht beilegen, sodass die Beispiele nur zum Herunterladen zur Verfügung stehen. Dies hat aber den Vorteil, dass die ständige Weiterentwicklung der Beispiele sich dort in Aktualisierungen niederschlägt.
- **Diskussionsrunde:** Ein webbasiertes Forum bietet die Möglichkeit, Fragen an den Autor zu stellen. Bitte beachten Sie jedoch, dass dies eine freiwillige Leistung des Autors ist und kein Anspruch auf eine kostenlose Betreuung besteht.
- **Newsletter:** Alle registrierten Leser erhalten mehrmals jährlich einen Newsletter mit aktuellen Terminen und Publikationshinweisen.
- **Leser-Bewertung:** Vergeben Sie Noten für dieses Buch und lesen Sie nach, was andere Leser von diesem Buch halten.
- **Errata:** Trotz eines erprobten Vorgehensmodells und der mehrfachen Qualitätskontrolle ist es möglich, dass sich einzelne Fehler in dieses Buch eingeschlichen haben. Im Webportal können Sie nachlesen, welche Fehler gefunden wurden. Sie können hier auch selbst Fehler melden, die Ihnen auffallen.

Zugang zum Leser-Portal

Der URL für den Zugang zum Leser-Portal lautet:

<http://www.dotnetframework.de/leser>

Bei der Anmeldung müssen Sie das Kennwort *Farscape* angeben.

Themeneinschränkungen

Das .NET Framework ist ein komplexes Programmiermodell, dessen komplette Beschreibung mehrere Tausend Seiten umfassen würde. Einige Aspekte des .NET Framework kann dieses Buch aufgrund der Seitenrestriktion nicht behandeln.

An dieser Stelle sei auf einige Themen explizit hingewiesen, die in diesem Buch aus Platzgründen nicht behandelt werden:

- C++/CLI, JScript.NET, IronPython und alle .NET-Programmiersprachen, die weniger verbreitet sind als C# und Visual Basic .NET
- Team Foundation Server (TFS)
- Programmierung mobiler Endgeräte mit dem .NET Compact Framework (CF)
- Programmierung sehr kleiner Endgeräte mit dem .NET Micro Framework (MF)
- Entwicklung von Windows-Diensten
- Analyse von Code und dynamische Code-Erzeugung mit System.Reflection
- Multi-Threading mit System.Threading
- Tracing mit System.Diagnostics
- Manuelle Nutzung der Serialisierer in .NET (BinaryFormatter, SoapFormatter, XmlSerializer, DataContractSerializer, NetDataContractSerializer) unabhängig von Webservices, .NET Remoting und WCF
- Berichterstellung mit Crystal Reports for Visual Studio und SQL Server Reporting Services
- Kryptografische Mechanismen
- Code Access Security (CAS)
- Windows CardSpaces (WCS)
- .NET Platform Invoke (P-Invoke) zum Aufruf von C-Bibliotheken
- Message Queuing mit MSMQ und System.Messaging
- Erstellung von MSI-Setup-Paketen
- Visual Studio Tools for Microsoft Office (VSTO) zur Entwicklung von Office-Anwendungen
- Programmierung mit .NET innerhalb des Microsoft SQL Server 2005 (SQLCLR)
- .NET-Programmierung in Microsoft SharePoint
- Microsoft Biztalk Server

...und vieles mehr.

Ihre Entwicklungsumgebung für .NET

Diesem Buch liegt eine DVD mit einer Testversion von Visual Studio 2005 Team Suite bei. Durch das Installationsprogramm wird Ihr Rechner mit vielem ausgestattet, was zum komfortablen Entwickeln von .NET-Anwendungen notwendig ist. Dies sind im Wesentlichen drei Bausteine:

1. .NET Framework 2.0 Redistributable
2. .NET Framework Software Development Kit (SDK)
3. Entwicklungsumgebung Visual Studio 2005

Für .NET 3.0 müssen Sie sich aus dem Internet außerdem besorgen:

1. .NET Framework 3.0 Redistributable
2. Windows SDK 5.0
3. Visual Studio 2005 Erweiterungen für Windows Workflow Foundation
4. Visual Studio 2005 Erweiterungen für Windows Presentation Foundation und Windows Communication Foundation (diese Erweiterung gibt es zum Redaktionsschluss des Buches erst in einer frühen, fehlerbehafteten Vorabversion)

Microsoft hat die Lizenz für die Testversion von Visual Studio auf 180 Tage beschränkt. An dieser Stelle möchte der Autor Sie darüber informieren, welche Möglichkeiten Ihnen nach Ablauf des Testzeitraums zur Verfügung stehen.

Redistributable und SDK

Um .NET-Anwendungen zu entwickeln, brauchen Sie von den drei oben genannten Bausteinen eigentlich nur das Redistributable und das SDK. Da diese beiden Bausteine kostenlos sind, gilt die Zeitbeschränkung dafür nicht. Das heißt, dass das Redistributable und das SDK auch nach Ablauf der 180 Tage noch funktionieren werden. Lediglich die Entwicklungsumgebung Visual Studio 2005 wird nicht mehr benutzbar sein.

Sie können das .NET Framework Redistributable und das SDK auch separat von der DVD installieren (siehe Verzeichnisse /WCU/dotnetframework und /WCU/SDK) oder von der Microsoft Homepage herunterladen:

- <http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

und

- <http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

Tatsächlich sind Redistributable und das SDK die Mindestvoraussetzungen. In der professionellen Anwendungsentwicklung wird man darüber hinaus eine Entwicklungsumgebung einsetzen wollen.

Entwicklungsumgebung Visual Studio

Grundsätzlich liegt es natürlich im geschäftlichen Interesse der Firma Microsoft, dass Sie nach Ablauf des Testzeitraums eine kostenpflichtige Lizenz der Entwicklungsumgebung Visual Studio 2005 erwerben. Der Autor möchte Sie in Ihrer Entscheidung diesbezüglich nicht beeinflussen, sondern Ihnen an dieser Stelle einen Weg aufzeigen, wie Sie auch ohne Kosten für die Entwicklungsumgebung weiterhin komfortabel .NET-Anwendungen entwickeln können.

Express-Editionen

Microsoft bietet neben Visual Studio 2005 auch noch eine Produktfamilie unter dem Titel »Visual Studio Express Editionen« an. Bei den Express-Editionen handelt es sich um größere Bausteine, die aus Visual Studio herausgebrochen wurden und jeweils für einen speziellen Anwendungsfall bereitstehen. Diese Express-Editionen sind kostenlos auf der Microsoft Website beziehbar. Sie dürfen ohne Gebühren unbegrenzt verwendet werden. Es existieren derzeit die in nachfolgender Tabelle genannten Express-Editionen. Die Express-Editionen reichen für die wichtigsten Anwendungstypen aus. Es gibt keine Express-Editionen für exotischere .NET-Anwendungsarten, z.B. Microsoft Office-Anwendungen oder SQL Server-Anwendungen.

	Programmiersprachen	Konsolen-Anwendungen	Windows-Anwendung	Bibliotheken (DLL)	Web-Anwendungen	Webservices
C++ Express	C++ 8.0	X	X	X		X (WCF)
C# Express	C# 2005	X	X	X		X (WCF)
Visual Basic Express	Visual Basic 2005	X	X	X		X (WCF)
J# Express	J# 2005	X	X	X		X (WCF)
Visual Web Developer Express	C# 2005 und Visual Basic 2005				X	X (ASMX)

Tabelle 0.1 Übersicht über die Express-Editionen von Visual Studio

Die Express-Editionen können Sie von folgender Website beziehen:

<http://msdn.microsoft.com/vstudio/express/default.aspx>

Installation der Express-Editionen

Sie können alle Express-Editionen problemlos gemeinsam auf einem Computersystem installieren. Jedoch integrieren sich die Produkte nicht ineinander, d.h. alle Entwicklungsumgebungen besitzen ein eigenes Anwendungsfenster, auch eine Mischung von verschiedenen Projekttypen in einer Projektmappe ist nicht möglich. Diese Integration hat Microsoft den käuflich zu erwerbenden Visual Studio 2005-Vollprodukten vorbehalten.

Im Dateisystem verwenden die fünf Express-Editionen viele Dateien gemeinsam, sodass der Platzbedarf der Produkte zusammen weniger ist als die Summe der Einzelinstallationen. Alle Express-Editionen enthalten auch den SQL Server 2005 Express Edition und eine reduzierte Version der MSDN-Dokumentation. Diese Zusatzwerkzeuge werden nur bei der Installation der ersten Express-Edition auf die Festplatte gespielt.

Die Express-Editionen müssen nicht bei Microsoft aktiviert oder registriert werden. Die optionale, kostenlose Registrierung verbindet Microsoft jedoch mit einigen Angeboten. Wer sich registriert, erhält Zugriff auf lizenzfreie Bilder und Symbole, E-Books und Softwarekomponenten.

Hinweise zu den Listings

Dieses Buch orientiert sich an einem übergreifenden Fallbeispiel, der Fluggesellschaft »World Wide Wings« (siehe Kapitel 2). Während viele Programmcodebeispiele direkt in die entsprechenden Anwendungsteile des Gesamtbeispiels integriert sind, liegen aus didaktischen Gründen einige Beispiele aus den Kapiteln »Visual Basic 2005 (VB 8.0)«, »C# 2005 (C# 8.0)« und »NET-Klassenbibliothek 2.0« in autonomen, besser überschaubaren Codebeispielen vor. Diese Codebeispiele verwenden zwar oft auch Daten aus dem Fallbeispiel, sind jedoch einzeln startbar. Zum Start dieser autonomen Beispiele wird eine von dem Autor dieses Buches entwickelte Rahmenanwendung, der *Demo-Viewer* eingesetzt. Der Demo-Viewer erlaubt die Anzeige und den Start aller Beispiele, die sich in einer definierbaren Assembly befinden.

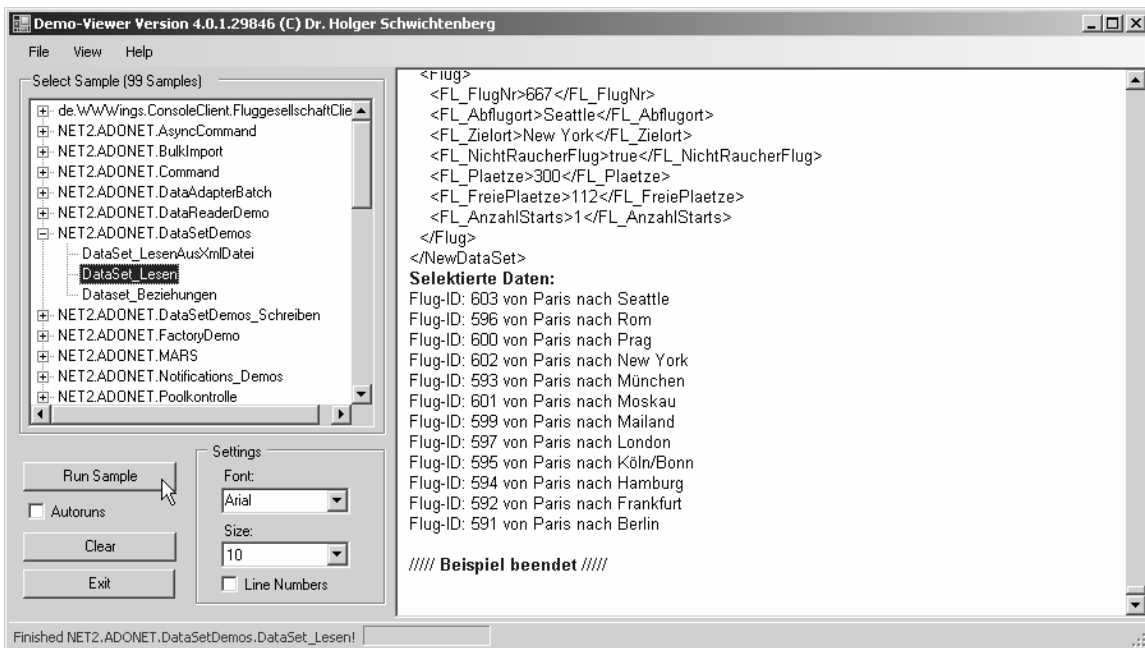


Abbildung 0.1 Start eines Beispiels im Demo-Viewer

Sie werden in den herunterladbaren Beispielen zu diesem Buch bzw. in den im Buch abgedruckten Listings folgende Vorkehrungen zur Nutzung des Demo-Viewers finden:

- Projekte, die Beispiele enthalten, referenzieren folgende Softwarekomponente: *ITV_DemoViewer.dll*
- Codedateien, die Beispiele enthalten, importieren folgenden Namensraum: `de.ITVisions.DemoViewer`
- Klassen, die Beispiele enthalten, sind mit `[Demo]` bzw. `<Demo>` annotiert.
- Ausführbare Beispiele sind Instanzmethoden und besitzen keine Parameter.
- Ausgaben in den Beispielen erfolgen mit `Demo.Print()` – für normale Ausgaben – und `Demo.PrintHeader()` – für hervorgehobene Ausgaben.
- Die Anwendung startet den Demo-Viewer für C# mit:

```
new de.ITVisions.DemoViewer.Viewer(System.Reflection.Assembly.GetExecutingAssembly())
```

bzw. für Visual Basic mit:

```
Dim d As New Viewer(System.Reflection.Assembly.GetExecutingAssembly())
```

Wenn Sie Beispiele aus diesem Buch in eigene Anwendungen übernehmen wollen, müssen Sie lediglich die o.g. Verbindungen zu dem Demo-Viewer entfernen, insbesondere die Ausgaben mit `Demo.Print()` und `Demo.PrintHeader()` ersetzen (siehe Tabelle).

Anwendungsform	Einfache sequenzielle Ausgabe	Ausgabe in bestimmtes Element
Konsolenanwendung	<code>Console.WriteLine(Ausgabe)</code>	Nicht möglich
ASP.NET-Web Forms-Anwendungen	<code>Response.Write(Ausgabe)</code>	<code>TextBox1.Text += Ausgabe</code>
Windows Forms-Anwendung	Nicht möglich	<code>RichTextBox1.AppendText(Ausgabe)</code>

Tabelle 0.2 Alternative Ausgabemöglichkeiten