

Inhaltsverzeichnis

Danksagung.....	17
Einführung.....	19
An wen sich das Buch richtet.....	20
Wie das Buch organisiert ist	20
Systemanforderungen.....	20
SQL Server 2005 Express Edition konfigurieren	21
Codebeispiele	22
Support.....	22
Teil A – Erste Schritte mit Microsoft ADO.NET.....	25
1 Überblick über ADO.NET	27
Kein neues Objektmodell?!?	28
Das ADO.NET-Objektmodell	29
.NET-Datenprovider.....	30
Warum werden separate Klassen und Bibliotheken verwendet?	32
Die Behandlung der .NET-Datenprovider in diesem Buch	33
Verbundene Objekte.....	34
Nicht verbundene Klassen	37
Metadaten.....	44
Stark typisierte <i>DataSet</i> -Klassen	45
Fragen, die häufiger gestellt werden sollten.....	46
2 Ihre erste ADO.NET-Anwendung mit Microsoft Visual Studio 2005 erstellen	49
Jeder weiß eine Demo zu schätzen	50
Der DataForm-Assistent wurde verworfen.....	50
Das Datenzugriffsformular ohne Code erstellen.....	51
Das neue Projekt erstellen.....	51
Eine neue Datenquelle hinzufügen.....	52
Die Datenquelle verwenden, um Elemente in das Formular einzufügen.....	67
Das Projekt ausführen	69
Verknüpfte Daten in das Formular einfügen.....	71
Den von Visual Studio generierten Code untersuchen	73
Besseren Datenzugriffscode als die Designer schreiben!.....	75
Fragen, die häufiger gestellt werden sollten.....	76

Teil B – Verbindungen herstellen – einen .NET-Datenprovider verwenden	79
3 Verbinden mit Ihrer Datenbank.....	81
<i>SqlConnection</i> -Objekte erzeugen.....	82
<i>SqlConnection</i> -Objekte öffnen	83
<i>SqlConnection</i> -Objekte schließen	83
Bereinigen in eigener Regie	84
Verbindungszeichenfolgen	87
Was ist eine Verbindungszeichenfolge?	87
Builder für Verbindungszeichenfolgen einführen	89
Sicherheit von Verbindungszeichenfolgen	96
Verbindungspooling.....	100
Verbindungshandles und physische Verbindungen	101
Was ist Verbindungspooling?	101
Wie Verbindungspooling Ihren Code verbessern kann	103
Verbindungspooling aktivieren.....	103
Wann wird die im Pool liegende Verbindung geschlossen?	104
Verbindungspooling deaktivieren.....	104
Beantworten Ihrer eigenen Fragen zum Verbindungspooling	105
Wie ADO.NET ermittelt, ob eine Verbindung aus dem Pool verwendet wird.....	106
ADO.NET dazu zwingen, einen neuen Pool zu verwenden.....	106
Gepoolte Verbindungen manuell freigeben	107
Welche anderen Poolingoptionen habe ich?	107
<i>SqlConnection</i> als Ausgangspunkt verwenden.....	108
<i>SqlCommand</i> -Objekte erzeugen.....	109
<i>SqlTransactions</i> starten	109
Schemainformationen abrufen	110
Entwicklungszeitfeatures von Visual Studio	117
Arbeiten mit Verbindungen in Server-Explorer	117
Eine Datenverbindung zu Server-Explorer hinzufügen.....	119
Was ist mit den Drag&Drop-Features passiert?	121
Klassenreferenz für <i>SqlConnection</i>	121
Eigenschaften der Klasse <i>SqlConnection</i>	121
Methoden der Klasse <i>SqlConnection</i>	125
Ereignisse der Klasse <i>SqlConnection</i>	133
Fragen, die häufiger gestellt werden sollten	135
4 Ihre Datenbank abfragen.....	139
<i>SqlCommand</i> -Objekte im Code verwenden	140
Ein <i>SqlCommand</i> -Objekt erzeugen.....	140
Abfragen mit einem <i>SqlCommand</i> ausführen	141
Eine Abfrage, die Zeilen zurückgibt, ausführen.....	141
Einen einzelnen Wert abrufen	144
Eine Abfrage ausführen, die kein Resultset zurückgibt	146

Aktionsabfragen im Batch ausführen	149
Eine Abfrage ausführen, um XML-Daten abzurufen	151
Eine Abfrage in einer Transaktion ausführen	151
Eine Abfrage asynchron ausführen	152
Mit dem <i>SqlDataReader</i> arbeiten	159
Die Ergebnisse Ihrer Abfrage untersuchen	160
Den <i>SqlDataReader</i> schließen	161
Das Schema des Resultsets untersuchen	163
Daten mit ordinalbasiertem Suchen schneller abrufen	167
Stark typisierte Getter	168
NULL-Werte behandeln	169
<i>SqlTypes</i>	172
Mehrere Resultsets aus einer Abfrage behandeln	174
SQL Server 2005 und mehrere aktive Resultsets	175
Mit parametrisierten Abfragen arbeiten	180
Benutzereingaben in einer Abfragezeichenfolge formatieren	180
Abfragekonstruktion und SQL-Injection	181
Parametrisierte Abfragen	182
Gespeicherte Prozeduren	187
Entwurfszeitfeatures von Microsoft Visual Studio	188
Was ist mit den Drag&Drop-Features passiert?	190
Klassenreferenz für <i>SqlCommand</i> , <i>SqlDataReader</i> und <i>SqlParameter</i>	190
Eigenschaften der Klasse <i>SqlCommand</i>	190
Methoden der Klasse <i>SqlCommand</i>	193
Ereignis der Klasse <i>SqlCommand</i>	198
Eigenschaften der Klasse <i>SqlDataReader</i>	198
Methoden der Klasse <i>SqlDataReader</i>	199
<i>SqlParameter</i> -Objekte erzeugen	204
Eigenschaften der Klasse <i>SqlParameter</i>	205
Fragen, die häufiger gestellt werden sollten	208
 5 Daten mit <i>SqlDataAdapter</i> -Objekten abrufen	 211
Was ist ein <i>SqlDataAdapter</i> -Objekt?	214
Wie sich der <i>SqlDataAdapter</i> von anderen Abfrageobjekten unterscheidet	215
Anatomie der Klasse <i>SqlDataAdapter</i>	218
<i>SqlDataAdapter</i> -Objekte erzeugen und verwenden	220
Einen <i>SqlDataAdapter</i> erzeugen	220
Abfrageergebnisse abrufen	223
Die Abfrageergebnisse dem <i>DataSet</i> zuordnen	229
Mit Batchabfragen arbeiten	231
Zeilen von einer gespeicherten Prozedur abrufen	233
Schemainformationen abrufen	233

Entwurfszeitfeatures von Visual Studio 2005	234
Was ist mit den Drag&Drop-Features passiert?	235
<i>SqlDataAdapter</i> -Klassenreferenz	235
Eigenschaften der Klasse <i>SqlDataAdapter</i>	236
Methoden der Klasse <i>SqlDataAdapter</i>	241
Ereignisse der Klasse <i>SqlDataAdapter</i>	248
Fragen, die häufiger gestellt werden sollten	253
Teil C – Mit Daten offline arbeiten – das ADO.NET- <i>DataSet</i>	255
6 Mit <i>DataSet</i> -Objekten arbeiten	257
Features der Klasse <i>DataSet</i>	258
Mit unverbundenen Daten arbeiten	258
Scrollen, sortieren, suchen und filtern	259
Mit hierarchischen Daten arbeiten	259
Änderungen zwischenspeichern	260
XML-Integration	260
Einheitliche Funktionalität	260
<i>DataSet</i> -Objekte verwenden	261
Ein <i>DataSet</i> -Objekt erzeugen	261
Die Struktur untersuchen, die durch den Aufruf von <i>SqlDataAdapter.Fill</i> erzeugt wurde	261
Die Daten untersuchen, die von einem <i>SqlDataAdapter</i> zurückgegeben wurden	263
Daten in Ihrem <i>DataSet</i> überprüfen	266
<i>DataTable</i> -Objekte im Code erzeugen	270
Autoinkrement-Gebote und -Verbote	278
Den Inhalt einer <i>DataTable</i> ändern	285
ADO.NET 2.0-Optionen für <i>DataSet</i> -Serialisierung und Remoting	297
Mit <i>DataSet</i> -Objekten in Visual Studio arbeiten	304
Stark typisierte <i>DataSets</i> in Visual Studio erzeugen	304
Ein nicht typisiertes <i>DataSet</i> erzeugen	305
<i>DataSet</i> -, <i>DataTable</i> -, <i>DataColumn</i> -, <i>DataRow</i> -, <i>UniqueConstraint</i> - und <i>ForeignKeyConstraint</i> -Klassenreferenz	309
Eigenschaften der Klasse <i>DataSet</i>	309
Methoden der Klasse <i>DataSet</i>	314
Ereignisse der Klasse <i>DataSet</i>	322
Das Ereignis <i>MergeFailed</i>	322
Eigenschaften der Klasse <i>DataTable</i>	323
Methoden der Klasse <i>DataTable</i>	328
Ereignisse der Klasse <i>DataTable</i>	335
Eigenschaften der Klasse <i>DataColumn</i>	338
Eigenschaften der Klasse <i>DataRow</i>	344
Methoden der Klasse <i>DataRow</i>	346

Eigenschaften der Klasse <i>UniqueConstraint</i>	353
Eigenschaften der Klasse <i>ForeignKeyConstraint</i>	354
Fragen, die häufiger gestellt werden sollten	356
7 Mit relationalen Daten arbeiten	363
Ein kurzer Überblick über relationalen Datenzugriff.....	365
Join-Abfragen	365
Separate Abfragen	366
Hierarchische ADO- <i>Recordset</i> -Objekte.....	367
ADO.NET- <i>DataRelation</i> -Objekte	368
Mit <i>DataRelation</i> -Objekten im Code arbeiten	369
<i>DataRelation</i> -Objekte erzeugen	369
In Beziehung stehende Daten suchen	372
Die Daten mit <i>DataRelation</i> -Objekten überprüfen	378
Auf sich selbst verweisende <i>DataRelationship</i> -Objekte.....	382
m:n-Beziehungen	385
<i>DataRelation</i> -Objekte in ausdrucksbasierten <i>DataColumn</i> -Objekten verwenden	388
Änderungen weitergeben	392
Von Join-Abfragen abrücken	393
<i>DataRelation</i> -Objekte in Microsoft Visual Studio erzeugen	394
Eine <i>DataRelation</i> zu einem stark typisierten <i>DataSet</i> hinzufügen.....	394
Eine <i>DataRelation</i> zu einem nicht typisierten <i>DataSet</i> hinzufügen	394
<i>DataRelation</i> -Klassenreferenz	395
Eigenschaften der Klasse <i>DataRelation</i>	395
Fragen, die häufiger gestellt werden sollten	399
8 Sortieren, suchen und filtern	405
Die Such- und Filterfeatures der Klasse <i>DataTable</i> verwenden.....	406
Eine Zeile nach ihren Primärschlüsselwerten suchen	406
Dynamischere Suchvorgänge durchführen.....	408
Suchvorgänge mit Platzhaltern durchführen	409
Mit Trennzeichen arbeiten	410
Die zusätzlichen <i>Select</i> -Methoden verwenden.....	412
Was ist ein <i>DataRowView</i> -Objekt?	415
<i>DataRowView</i> -Objekte geben Daten aus einer <i>DataTable</i> zurück.....	416
<i>DataRowView</i> -Objekte sind keine SQL-Abfragen.....	416
Mit <i>DataRowView</i> -Objekten im Code arbeiten	418
<i>DataRowView</i> -Objekte erzeugen.....	418
Die Eigenschaft <i>RowStateFilter</i> verwenden	420
Die Klasse <i>DataRowView</i> verwenden.....	420
Alle Datenzeilen untersuchen, die über eine <i>DataRowView</i> verfügbar sind	423
In einer <i>DataRowView</i> nach Daten suchen	424
<i>DataRowView</i> -Objekte ändern	426

Mit einer <i>DataGridView</i> eine neue <i>DataTable</i> erzeugen	427
<i>DataGridView</i> -Objekte in Microsoft Visual Studio erzeugen	431
Ein neues <i>DataGridView</i> -Objekt in Ihren Designer hinzufügen	431
Eigenschaften Ihres <i>DataGridView</i> -Objekts festlegen	431
<i>DataGridView</i> -Klassenreferenz	432
Methoden der Klasse <i>DataGridView</i>	435
Ereignisse der Klasse <i>DataGridView</i>	439
Eigenschaften der Klasse <i>DataRowView</i>	440
Methoden der Klasse <i>DataRowView</i>	441
Fragen, die häufiger gestellt werden sollten	444
9 Mit stark typisierten <i>DataSet</i> -Objekten und <i>TableAdapters</i> arbeiten	451
Stark typisierte <i>DataSets</i>	452
Stark typisierte <i>DataSet</i> -Objekte erzeugen	453
Der schwere Weg	454
Der einfache Weg	456
Stark typisierte <i>DataSet</i> -Objekte verwenden	462
Eine Zeile hinzufügen	462
Eine Zeile suchen	465
Eine Zeile bearbeiten	466
Mit NULL-Werten arbeiten	467
Mit hierarchischen Daten arbeiten	468
Andere <i>DataSet</i> -, <i>DataTable</i> - und <i>DataRow</i> -Features	469
Eigenen Code hinzufügen	469
Wann verwendet man stark typisierte <i>DataSet</i> -Objekte?	473
Softwarekomponenten und Taschenmesser	473
Entwurfszeitvorteile	474
Laufzeitvorteile	475
Weitere Überlegungen	481
Tabellen und Spalten manuell hinzufügen	484
Standardwerte für <i>DataSets</i> verbessern	487
Einführung in <i>TableAdapters</i>	488
Einen <i>TableAdapter</i> erzeugen	488
Einen <i>TableAdapter</i> verwenden	494
Weitere Abfragen hinzufügen	500
Eigenen Code hinzufügen	503
Beschränkungen von <i>TableAdapter</i>	503
Wählen Sie Ihren Weg	506
Fragen, die häufiger gestellt werden sollten	506
10 Aktualisierungen an Ihre Datenbank übermitteln	509
Aktualisierungen mit parametrisierten <i>SqlCommand</i> -Objekten übermitteln	516
Eine neue Zeile übermitteln	516

Eine vorhandene Zeile aktualisieren	517
Eine vorhandene Zeile löschen.....	525
Was brauchen Sie noch?.....	526
Aktualisierungen mit einem <i>SqlDataAdapter</i> übermitteln	527
Aktualisierungen mit <i>SqlDataAdapter</i> -Objekten übermitteln.....	530
Ihre <i>SqlDataAdapter</i> -Objekte manuell konfigurieren.....	530
Gebundene Parameter	530
Aktualisierungen mit gespeicherten Prozeduren übermitteln	533
Ihre eigene Aktualisierungslogik angeben.....	537
Mit dem <i>SqlCommandBuilder</i> Aktualisierungslogik generieren	538
Wie der <i>SqlCommandBuilder</i> Aktualisierungslogik generiert	539
Parallelitätsoptionen bei der Verwendung des <i>SqlCommandBuilder</i>	541
Vor- und Nachteile der Verwendung des <i>SqlCommandBuilder</i>	541
Mit dem <i>TableAdapter</i> -Konfigurations-Assistenten von Visual Studio Aktualisierungslogik erzeugen.....	542
Die Aktualisierungslogik des <i>TableAdapter</i> untersuchen.....	543
Optionen zum Erstellen von Aktualisierungslogik.....	544
Aktualisierungen mit gespeicherten Prozeduren übermitteln	545
Vor- und Nachteile des Assistenten	549
Rückkehr der <i>DataAdapter</i> !.....	549
Den <i>DataAdapter</i> -Konfigurations-Assistenten aufrufen.....	550
Eine einfache Datenzugriffsschicht mit <i>SqlDataAdapter</i> erzeugen.....	550
Aktualisierungen in <i>SqlTransactions</i> übermitteln	558
Die TableMappings-Auflistung verwenden	563
Die beste Methode der Aktualisierung	565
<i>SqlCommandBuilder</i> -Klassenreferenz.....	566
Eigenschaften der Klasse <i>SqlCommandBuilder</i>	566
Methoden der Klasse <i>SqlCommandBuilder</i>	568
Fragen, die häufiger gestellt werden sollten	571
11 Erweiterte Aktualisierungsszenarios.....	579
Eine Zeile nach der Übermittlung einer Änderung aktualisieren.....	580
Den Inhalt einer <i>DataRow</i> aktualisieren, nachdem Sie eine Änderung übermittelt haben	581
Mit Batchabfragen Daten abrufen, nachdem Sie eine Aktualisierung übermittelt haben	582
Neue Daten mit Ausgabeparametern abrufen.....	585
Mit dem <i>RowUpdated</i> -Ereignis der <i>SqlDataAdapter</i> -Klasse Daten abrufen, nachdem Sie eine Aktualisierung übermittelt haben	586
<i>RefreshAfterUpdate</i> -Beispielcode.....	589
Neu generierte Autoinkrementwerte abrufen	589
Mit SQL Server arbeiten.....	589
Mit Access-Datenbanken arbeiten.....	592
Mit Oracle-Sequenzen arbeiten.....	593

Beispielanwendungen, die Autoinkrementwerte abrufen	595
Hierarchische Änderungen übermitteln.....	596
Anstehende Einfügungen und Löschungen übermitteln.....	596
Mit Autoinkrementwerten und relationalen Daten arbeiten.....	599
Änderungen isolieren und erneut integrieren.....	601
Mit der Methode <i>GetChanges</i> Bandbreite sparen	601
Änderungen isolieren und erneut integrieren – Beispielcode	612
Fehlgeschlagene Aktualisierungsversuche behandeln	612
Konflikte einkalkulieren	613
Den Benutzer über Fehlschläge informieren.....	613
Den aktuellen Inhalt von widersprüchlichen Zeilen abrufen	615
Wenn Sie beim ersten Mal keinen Erfolg haben	616
Die Konflikte – Beispielanwendung.....	618
Mit verteilten Transaktionen arbeiten	618
Transaktionskoordinatoren und Ressourcenmanager.....	620
Verteilte Transaktionen im .NET Framework	621
Datenbankunterstützung für verteilte Transaktionen	621
Ihre Komponenten erstellen	622
System.Transactions – Beispiele.....	625
Batchabfragen.....	625
Transaktionen mit Batchaktualisierungen verwenden	626
Einen geeigneten Wert für <i>UpdateBatchSize</i> wählen	627
Ereignisse	627
Zeilen aktualisieren.....	628
Batchaktualisierungen – Beispiele.....	628
SQL Bulk Copy.....	628
Ein <i>SqlBulkCopy</i> -Objekt erzeugen	629
Daten zum Server schreiben.....	630
Daten auf die Zieltabelle abbilden.....	631
Die Enumeration <i>SqlBulkCopyOptions</i>	631
<i>LBulkCopy</i> -Beispielcode	632
<i>DataSet</i> -Objekte und Transaktionen	632
<i>DataSet</i> -Objekte und Transaktionen – Beispielcode	633
Verwenden Sie für erweiterte Aktualisierungsszenarios ADO.NET	633
Fragen, die häufiger gestellt werden sollten	633
 12 Mit XML-Daten arbeiten	 637
Die Lücke zwischen XML und dem Datenzugriff überbrücken	638
XML-Daten lesen und schreiben.....	639
Die XML-Methoden der Klasse <i>DataSet</i>	639
Schemas ableiten	643
ADO.NET-Eigenschaften, die sich auf das Schema Ihres XML-Dokuments auswirken	644
Änderungen und XML-Dokumente zwischenspeichern.....	646
<i>DataSet</i> + <i>XmlDocument</i> = <i>XmlDataDocument</i>	648

Die Klasse <i>XmlDataDocument</i> verwenden.....	649
Auf das <i>DataSet</i> als XML-Dokument zugreifen	649
Aktualisierungen des XML-Dokuments zwischenspeichern	650
Die XML-Features von SQL Server 2005.....	652
Der XML-Typ von SQL Server 2005	652
Eine XPath-Abfrage ausführen, um Daten abzurufen.....	657
Abfrageergebnisse mithilfe von XQuery als XML abrufen	659
XML-Daten via <i>SELECT...FOR XML</i> aus SQL Server 2000 abrufen.....	662
Der SQL XML .NET-Datenprovider	666
Mit einem <i>SqlXmlCommand</i> Daten in ein <i>XmlDocument</i> laden	666
Mit einem <i>SqlXmlAdapter</i> Daten in ein <i>DataSet</i> laden	668
Mit Vorlagenabfragen arbeiten	669
Mit XPath-Abfragen arbeiten.....	673
Eine XSL-Transformation anwenden	675
Aktualisierungen übermitteln	676
Ein einfaches ADO.NET- und XML-Beispiel.....	680
Zwei Wege, ein Ziel.....	681
ADO.NET und XML: Ein nettes Paar	681
Fragen, die häufiger gestellt werden sollten	682
 Teil D – Mit ADO.NET effektive Anwendungen erstellen	 687
 13 Effektive Windows-Anwendungen erstellen	 689
Mit Datenbindung schnell eine Benutzeroberfläche erstellen	690
Schritt 1: Ihr stark typisiertes <i>DataSet</i> erzeugen.....	691
Schritt 2: Einfache datengebundene Steuerelemente dem Formular hinzufügen	693
Schritt 3: Daten abrufen	699
Schritt 4: Durch die Ergebnisse navigieren	702
Schritt 5: Elemente hinzufügen und löschen.....	704
Schritt 6: Änderungen übermitteln.....	707
Schritt 7: Edit-, Update- und Cancel-Schaltflächen ergänzen	709
Schritt 8: Untergeordnete Daten betrachten	711
Schritt 9: Ein zweites Formular an dieselbe Datenquelle binden.....	716
Schritt 10: Die Benutzeroberfläche verbessern	717
Schritt 11: Wenn etwas (ganz) richtig gemacht werden soll	724
Datenbindung – Zusammenfassung	725
Überlegungen zum Anwendungsdesign	725
Nur die Daten abrufen, die Sie wirklich benötigen.....	725
Aktualisierungsstrategien.....	725
Verbindungsstrategien	729
Mit BLOB-Daten arbeiten.....	731
Benutzeroberflächen, die auf der Leistung von ADO.NET aufbauen.....	737
Fragen, die häufiger gestellt werden sollten	737

14	Effektive Webanwendungen erstellen	741
	Eine kurze Einführung in Webanwendungen.....	742
	ASP.NET erleichtert die Erstellung von Webanwendungen.....	742
	Das Gute und das Schlechte der Zustandslosigkeit	743
	Vergesslicher Server, dummer Client	743
	Die Verbindung mit Ihrer Datenbank.....	744
	Verbindungen mit integrierter Sicherheit	744
	Mit Access-Datenbanken arbeiten	747
	Herausforderungen bei der Interaktion mit Datenbanken in ASP.NET 1.0.....	747
	Datenquellensteuerelemente	748
	Daten mithilfe eines <i>SqlDataSource</i> -Steuerelements anzeigen	749
	Features des <i>SqlDataSource</i> -Steuerelements	755
	Daten zwischen erneuten Datenbankabfragen zwischenspeichern.....	764
	Der Ansatz, keinen Zustand zu speichern	764
	Daten auf dem Client zwischenspeichern.....	765
	Zustandsverwaltung auf Ihrem Webserver.....	768
	Die SQL Server 2005 Notification Services.....	770
	Zustandsverwaltung in Ihrer Datenbank	778
	Richtlinien für die Zustandsverwaltung.....	779
	Paging.....	780
	Paging-Features des <i>GridView</i> -Steuerelements	780
	Paging-Features der <i>DataAdapter</i> -Klassen.....	782
	Abfragen erstellen, die eine Datenseite zurückgeben.....	783
	Seitenspezifische SQL Server 2005 Abfragen mit ROW_NUMBER erstellen.....	784
	Fragen, die häufiger gestellt werden sollten	790
15	CLR-Integration in SQL Server 2005	793
	Damals: SQL Server mit erweiterten gespeicherten Prozeduren erweitern	794
	Heute: SQL Server durch CLR-Integration erweitern	794
	SQL/CLR-Code mit Microsoft Visual Studio 2005 einfacher erstellen.....	796
	Eigenschaften für Ihr SQL Server-Projekt festlegen	796
	Das Projekt erstellen und bereitstellen	799
	Elemente in Ihr Projekt hinzufügen.....	799
	Das Projekt mit einem SQL-Skript testen	800
	Szenarios für SQL/CLR.....	800
	Eine Skalarfunktion zur Datenprüfung erstellen	800
	Eine Aggregatfunktion erstellen.....	803
	Die aktuelle Datenbank über die Kontextverbindung abfragen.....	807
	SQL/CLR-Tabellenwertfunktionen erstellen.....	810
	Abfrageergebnisse aus einer gespeicherten Prozedur zurückgeben.....	813
	Daten über Parameter von gespeicherten Prozeduren zurückgeben.....	817
	Einen SQL/CLR-benutzerdefinierten Typ erstellen.....	819
	Was macht einen SQL/CLR-benutzerdefinierten Typ aus?	819
	Methoden und Eigenschaften für den benutzerdefinierten Typ verfügbar machen	823

Den benutzerdefinierten Typ in einer Clientanwendung einsetzen.....	828
Zusammenfassung.....	832
Fragen, die häufiger gestellt werden sollten.....	832
Anhang A – Andere .NET-Datenprovider verwenden	837
Das Provider-Factory-Modell.....	838
Beschränkungen der allgemeinen ADO.NET Schnittstellen.....	838
Wie das Provider-Factory-Modell früheren Beschränkungen begegnet	839
Beschränkungen des Provider-Factory-Modells	845
Datenbankschemasuche.....	849
Der ODBC .NET-Datenprovider	853
Eine Verbindung zu Ihrer Datenbank mit einer <i>OdbcConnection</i> herstellen	853
Abfragen unter Verwendung eines <i>OdbcCommand</i> ausführen	854
Die Ergebnisse einer Abfrage mit einem <i>OdbcDataAdapter</i> abrufen	858
Datenbank-Schemainformationen abrufen	859
Der OLE DB .NET-Datenprovider	862
Eine Verbindung zu Ihrer Datenbank mit einer <i>OleDbConnection</i> herstellen	862
Abfragen unter Verwendung eines <i>OleDbCommand</i> ausführen	864
Die Ergebnisse einer Abfrage mit einem <i>OleDbDataAdapter</i> abrufen	867
Datenbank-Schemainformationen abrufen	868
Der Oracle Client .Net-Datenprovider.....	871
Eine Verbindung zu Ihrer Oracle-Datenbank mit einer <i>OracleConnection</i> herstellen ...	872
Abfragen unter Verwendung eines <i>OracleCommand</i> ausführen	872
Die Klasse <i>OracleDataAdapter</i>	881
Datenbank-Schemainformationen abrufen	890
Anhang B – Beispiele und Tools	893
.NET-Datenprovider – DSP	894
Der ADO.NET-Data Explorer	895
<i>AdapterSet</i>	896
Stichwortverzeichnis	899
Über den Autor	927