

William R. Vaughn

# ADO 2.5

VB-Datenbankprogrammierung für Profis

Galileo Press

# Inhalt

Einleitung 11

Arbeiten mit ADO 15

- 1.1 **Auswahl der »richtigen« ADO-Version 15**
  - 1.1.1 ADO 2.5 kommt auf den Markt 16
  - 1.2 **Die neuen Features von ADO 2.5 18**
    - 1.2.1 »Record«- und »Stream«-Objekte 18
    - 1.2.2 URLs anstelle Von Verbindungszeichenfolgen 19
    - 1.2.3 Spezielle Felder für Dokumentquellenprovider 19
    - 1.2.4 Neue Referenzthemen 19
- 1.3 **Der ADO Component Checker 19**
- 1.4 **Installieren von ADO 20**
- 1.5 **Bereitstellen von ADO 21**
  - 1.5.1 Die Installer von Visual Studio und Windows 23
- 1.6 **Weitere Hilfe zu ADO 24**

2 **Einführung in ADO 25**

- 2.1 **ADO und OLE DB 25**
- 2.2 **Das ADO-Objektmodell 28**

2 **Erstellen von ADO-Objekten 33**

- 3.1 **Instanziieren von ADO-Objekten 33**
- 3.2 **Instanziieren von ADO-Komponenten in Visual Basic Script 35**

Verbindungsherstellung 37

- 4.1 **Das ADO-»Connection«-Objekt 38**
- 4.2 **Verwenden des »Connection«-Objekts 41**
  - 4.2.1 Bereich des »Connection«-Objekts 41
  - 4.2.2 Setzen der »Provider«-Eigenschaft 42
  - 4.2.3 **Verbindungsherstellung zu systemeigenen OLE DB- und ODBC-Datenprovidern 43**
    - f
  - 4.2.4 Nachträgliches Öffnen von Verbindungszeichenfolgen 47
  - 4.2.5 Codieren minimalistischer Verbindungszeichenfolgen 54
  - 4.2.6 Setzen der »CursorLocation«-Eigenschaft 59
  - 4.2.7 Setzen der Eigenschaften für die Zeitüberschreitung 61
  - 4.2.8 Aufforderung zur Eingabe von Benutzer-ID und Kennwort 62

- 4.2.9 Auswahl der Standarddatenbank 66
- 4.2.10 Verwenden des Datenansichtfensters zum Erstellen von Verbindungen 69
- 4.2.11 Verwenden von Datenquellensteuerelementen 74
- 4.2.12 Ausführungsmethoden für Verbindungsabfragen 74
- 4.2.13 Verbindungspooling 75
- 4.2.14 Asynchrones Öffnen einer Verbindung 82
- 4.2.15 Prüfen, ob die Verbindung noch steht 83
- 4.3 Verwalten von Transaktionen 84**
  - 4.3.1 Ereignisse für das »Connection«-Objekt 85
  - 4.3.2 Sind Ereignisse in der mittleren Schicht sinnvoll? 88
  - 4.3.3 Pas »InfoMessage«-Ereignis 88
  - 4.3.4 Fehlerbehandlung 88

## ADO-Befehlsstrategien 91

- 5.1 **Das Innenleben des »Command«-Objekts 92**
  - 5.1.1 Substituieren von Parameterwerten 93
  - 5.1.2 Wiederverwenden von Ausführungsplänen 95
- 5.2 **Erstellen von »Command«-Objekten 95**
  - 5.2.1 Festlegen der ADO-Befehlseigenschaften 97
  - 5.2.2 Handhabung parameterbasierter Abfragen 107
  - 5.2.3 Codieren von »Command«-Objekten für gespeicherte Oracle-Prozeduren 124
- 5.3 **Verwalten von »Command«-Objekten 125**
  - 5.3.1 Ausführen und erneutes Ausführen von »Command«-Objekten 126
  - 5.3.2 Ablaufverfolgung für Server und Datenprovider 131
  - 5.3.3 Ermitteln des Befehlsstatus 134
  - 5.3.4 Die »Cancel«-Methode 135
  - 5.3.5 Plan B für ADO-»Command«-Objekte 135
  - 5.3.6 Tipps und Warnungen zum »Command«-Objekt 136
  - 5.3.7 Ereignisbehandlung für »Connection«-Objekte 137
- 5.4 **Überlegungen hinsichtlich der Leistung: Sinnvoller Einsatz von »Command«-Objekten 137**

## Recordsetstrategien 139

- 6.1 **Einführung in Recordsets 139**
  - 6.1.1 Inhalt eines Resultsets 141
  - 6.1.2 Entwicklungsstrategien 141
- 6.2 **Erstellen von »Recordset«-Objekten 145**
  - 6.2.1 Funktionsweise des »Field«-Objekts 146

<b>6.3</b>	<b>Arbeiten mit »Recordset«-Objekten</b>	<b>148</b>
6.3.1	Erstellen effizienter »Recordset«-Objekte	149
6.3.2	Verwenden der Methode »Recordset Open«	150
6.3.3	Einstellen weiterer »Recordset«-Eigenschaften vor Verwenden der »Open«-Methode	159
6.3.4	Verwenden der »Supports«-Methode	164
6.3.5	Verwenden der Methode »Recordset Clone«	166
6.3.6	Verwenden der Methode »Recordset Close«	166
6.3.7	Verwalten der Cursorsmitgliedschaft	167
6.3.8	Verwalten der aktuellen Zeile	174
6.3.9	Verwalten von Einfügevorgängen für das Recordset	176
6.3.10	Verwalten von Recordsetaktualisierungen	182
6.3.11	Übermitteln und Verwalten mehrerer Operationen	211
6.3.12	Abbrechen, Rückgängigmachen und Rollback	217
6.3.13	Verwenden der Methode »Recordset Save«	218

## **Bearbeiten des Recordsets 221**

7.1	<b>Binden von Daten an das Recordset</b>	<b>221</b>
7.2	<b>Methoden für den Datenabruf</b>	<b>222</b>
7.2.1	Anzeigen von Zeilen mit dem »MSHFlexGrid«-Steuerelement	223
7.2.2	Anzeigen von Zeilen mit Hilfe des Sofortfensters	225
7.2.3	Asynchroner Datenabruf	226
7.2.4	»Do Until RS.EOF«-Schleifen	227
7.2.5	Abrufen von Recordsets mit »GetRows« und »GetString«	228
7.2.6	Verwenden der »RecordCount«-Eigenschaft	230
7.2.7	Arbeiten mit »Variant«-Arrays	232
7.3	<b>Sortieren, Filtern und Auffinden</b>	<b>233</b>
7.3.1	Erstellen von Filterzeichenfolgen	237
7.4	<b>Trennen von Recordsets</b>	<b>238</b>
7.5	<b>RO-Recordsets und RW-Recordsets im Vergleich</b>	<b>239</b>
7.6	<b>Arbeiten mit »Field«-Objekten</b>	<b>239</b>
7.6.1	Die mysteriöse »DataFormat«-Eigenschaft	241
7.6.2	»Field«-Objektmethoden - Die Gebrüder Chunk	242
7.7	<b>Arbeiten mit gespeicherten Prozeduren</b>	<b>247</b>
7.7.1	Ausführen von gespeicherten Prozeduren	249
7.7.2	Handhabung der Resultsets von gespeicherten Prozeduren	252
7.7.3	Handhabung komplexer Resultsets	261
7.7.4	Handhabung der automatischen Parametereinfügung in Transaktionen	267
7.7.5	Bereitstellen der Parameter für die gespeicherte Prozedur per Hand	268
7.7.6	Verwendbare Parameter	271
7.8	<b>Zugreifen auf veraltete Recordsets</b>	<b>273</b>

	Übergeben von Resultsets zwischen Schichten	275
<b>8.1</b>	<b>Pizzalieferung per LKW</b>	<b>276</b>
8.1.1	Wenn »Keine Sardellen, bitte« nicht ausreicht	278
<b>8.2</b>	<b>Grundlegendes zum Marshaling von Recordsets</b>	<b>280</b>
8.2.1	Standardmäßiges ADO-Marshaling	280
8.2.2	Jenseits des Standardmarshaling	281
8.2.3	Beschränkungen der Recordsetübergabe als Wert	282
8.2.4	Prozessexternes Marshaling	282
8.2.5	Leistung beim Marshaling	284
<b>8.3</b>	<b>Übergeben nicht verbundener Recordsets</b>	<b>287</b>
8.3.1	Der Code für die Serverseite	288
8.3.2	Der Code für die Clientseite	289
<b>8.4</b>	<b>Übergeben von Zeichenfolgen und Zeichenfolgenarrays</b>	<b>294</b>
8.4.1	Auffüllen einer Tabelle mit einer Zeichenfolge im Clip-Format	294
<b>8.5</b>	<b>Übergabe von Ausgabeparametern anstelle von Rowsets</b>	<b>297</b>
8.5.1	Der Code für die Serverseite	297
8.5.2	Der Code für die Clientseite	300
<b>8.6</b>	<b>Übergabe von »Variant«-Arrays</b>	<b>301</b>
8.6.1	Der Code für die Serverseite	302
8.6.2	Der Code für die Clientseite	302
<b>8.7</b>	<b>Übergabe benutzerdefinierter Strukturen</b>	<b>303</b>
8.7.1	Der serverseitige UDT-Code	305
8.7.2	Der Code für die Clientseite	311
<b>8.8</b>	<b>Übergeben von Eigenschaftensammlungen</b>	<b>316</b>
8.8.1	Verwenden des »PropertyBag«-Objekts zur Datenübertragung	318
8.8.2	»PropertyBag«-Servercode	318
8.8.3	»PropertyBag«-Clientcode	319
<b>8.9</b>	<b>Übergabe von XML</b>	<b>320</b>
<b>9</b>	<b>Webbasierte Lösungen</b>	<b>323</b>
<b>9.1</b>	<b>Webentwicklung</b>	<b>101 323</b>
9.1.1	Fehlerbehandlung in VBScript	325
9.1.2	Der HTML-Interpreter des Browsers	327
9.1.3	Ausführen von ActiveX Server Pages	328
9.1.4	Angaben von Parametern	330
9.1.5	Visual Basic im Vergleich zur VBScript-Entwicklung	331
<b>9.2</b>	<b>Verwenden von XML zum Zurückgeben von Daten aus Webseiten</b>	<b>331</b>
9.2.1	Die serverseitige Active Server Page	332
<b>9.3</b>	<b>Verwalten von Recordsets mit Hilfe von XML-»Stream«-Objekten</b>	<b>338</b>
9.3.1	Untersuchen des Codebeispiels von Visual Basic zu ASP	340
9.3.2	Der clientseitige Visual Basic-Code	341
9.3.3	Der serverseitige ASP-Abfragecode	343
9.3.4	Der serverseitige ASP-Aktualisierungscode	345

<b>10</b>	<b>Die Daten in Form bringen</b>	<b>349</b>
10.1	Der Shapeprovider	349
10.2	Erstellen einer einfachen »Shape«-Anweisung	352
10.3	Warum nicht den Datenumgebungs-Designer verwenden?	358
10.4	Analysieren der Auswirkungen	361
10.5	Auf der Suche nach einer besseren Lösung	363
10.6	Die Auswirkungen des Bedarfsansatzes	364
<b>11</b>	<b>Tipps und Vorgehensweisen für den Datenzugriff</b>	<b>367</b>
11.1	Referenziell von Daten auf anderen Servern	367
11.2	TSQL-Tipps	368 *
11.2.1	Verwenden der TOP-Klausel	369
11.2.2	Verwenden von SELECT INTO anstelle roher Gewalt	370
11.2.3	Verwenden von UNION zum Kombinieren identischer Recordsets	371
11.2.4	Zuweisen von und Prüfen auf NULL	371
11.3	Leistungsoptimierung für gespeicherte Prozeduren	371
11.4	Verwenden von SET NOCOUNT über den OLE DB-Provider	372
11.5	Gewähren von Berechtigungen	373
11.6	Weitere Geheimnisse über Recordsets	375
11.6.1	Erhöhen der Recordsetleistung	377
11.6.2	Optimieren der Filterkriterien	378
11.6.3	Fehlerbehandlung von der richtigen Quelle	378
11.6.4	Arbeiten mit dem »Schema«-Objekt	379
11.7	Arbeiten mit dem »Grid«-Steuerelement	380
11.8	Arbeiten mit Zeichenfolgen	381
11.8.1	Verwenden der Visual Basic-Operatoren für die Zeichenfolgenbearbeitung	381
11.8.2	Verschlüsseln von Zeichenfolgen	381
11.9	Arbeiten mit Grafiken	383
11.10	Probleme beim MDAC/ADO-Setup	383
11.11	Gemeinsame Anzeige von Visual Basic- und ADO-Hilfe	384
<b>12</b>	<b>ADO und die Visual Database Tools</b>	<b>385</b>
12.f	Verwenden des Datenansichtfensters	386
12.1.1	Erstellen von Datenlinks	386
12.1.2	Arbeiten mit dem Datenbankschema	387
12.1.3	Untersuchen von Tabellen, Sichten und gespeicherten Prozeduren	389
12.1.4	Was fehlt	391

<b>12.2</b>	<b>Verwenden des Datenumgebungs-Designers</b>	<b>392</b>
12.2.1	Datenumgebungs-Designer-grundlegende Funktionen	393
12.2.2	Fehlerbehandlung für Verbindungen im Datenumgebungs-Designer	397
12.2.3	Was im Datenumgebungs-Designer fehlt	398
<b>12.3</b>	<b>Verwenden von ADO ohne Datenumgebungs-Designer</b>	<b>401</b>
<b>12.4</b>	<b>Verwenden des Datenobjekt-Assistenten</b>	<b>401</b>
12.4.1	Phase 1 - Vorbereitung	402
12.4.2	Phase 2 - Erstellen der »Recordset«- und »DataSource«-Klassen	403
12.4.3	Phase 3 - Erstellen eines benutzerdefinierten Benutzersteuerelements	403
13	SQL Server 2000 »Jnd ADO 2.6	405
13.1	<b>SQL Server 2000 und der Anwendungsentwickler</b>	406
13.1.1	Skalierbarkeit und Leistung	406
13.1.2	MSDE-Änderungen	407
13.1.3	Mehrfachinstanzierung	408
13.1.4	TSQL-Erweiterungen	408
13.1.5	XML-Integration	410
13.2	<b>ADO 2.6</b>	4«
13.2.1	SQL-XML	412
13.2.2	ODBC-Entwicklung	413
	Anhang	415
	CursorLocation: serverseitige Cursor	415
Index		419