

Leseprobe

Walter Doberenz, Thomas Gewinnus

Access programmieren

Grundlagen und Praxiswissen. Für die Versionen 2010, 2013 und 2016

ISBN (Buch): 978-3-446-45027-1

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-45027-1>

sowie im Buchhandel.

Inhaltsverzeichnis

Vorwort	27
 Teil I: Grundlagen	
1 Einführung	33
1.1 VBA-Programmierung in Access	33
1.1.1 Visual Basic versus VBA	33
1.1.2 Objekt- und ereignisorientierte Programmierung	34
1.1.3 VBA- oder Makro-Programmierung?	35
1.1.4 Die VBA-Entwicklungsumgebung	36
1.1.5 Formularentwurf	36
1.1.6 Code-Fenster und Symbolleiste	37
1.1.7 Das Eigenschaftfenster	38
1.1.8 Der Projekt-Explorer	39
1.1.9 Das Code-Fenster	39
1.1.10 Arbeiten mit der Hilfe	42
1.2 Sicherheitseinstellungen	43
1.2.1 Zur Geschichte der Access-Sicherheit	43
1.2.2 Eine nicht vertrauenswürdige Datenbank öffnen	44
1.2.3 Das Sicherheitscenter	46
1.2.4 Definition vertrauenswürdiger Speicherorte	49
1.3 Einführungsbeispiele	50
1.3.1 Erstellen der Testdatenbank	51
1.3.2 Konventionelle Programmierung	52
1.3.3 Programmieren mit VBA	56
1.3.4 Automatische Makrokonvertierung	61
1.3.5 Programmieren mit Datenmakros	62
1.4 Highlights und Features von Access 2016	64
1.4.1 Zur Geschichte der Vorgängerversionen	64
1.4.2 Microsoft Access 2016 – viel Lärm um nichts?	67
1.4.3 Der inoffizielle Access-Friedhof (Access 2013/2016)	68

1.5	Übersichten und Ergänzungen	68
1.5.1	Deutsche und englische Bezeichner	68
1.5.2	DoCmd-Objekt	70
2	Programmieren mit VBA	73
2.1	Datentypen, Variablen und Konstanten	73
2.1.1	Übersicht	73
2.1.2	Variablendeklaration	74
2.1.3	Konstantendeklaration	79
2.1.4	Gültigkeitsbereiche	80
2.2	Einzelheiten zu den Datentypen	83
2.2.1	Single- und Double-Datentypen	83
2.2.2	Integer-, Long- und Boolean-Datentypen	83
2.2.3	Date-Datentyp	84
2.2.4	Currency-Datentyp	86
2.2.5	String-Datentyp	87
2.2.6	Variant-Datentyp	89
2.3	Datenfelder (Arrays)	92
2.3.1	Statische Arrays	92
2.3.2	Dynamische Arrays	94
2.4	Benutzerdefinierte Datentypen	96
2.4.1	Type-Anweisung	96
2.4.2	With-Anweisung	97
2.4.3	Strings innerhalb Type	97
2.4.4	Enumerationen	98
2.4.5	Arrays in benutzerdefinierten Typen	99
2.5	Operatoren	100
2.5.1	Arithmetische Operatoren	101
2.5.2	Logische Operatoren	103
2.5.3	Vergleichsoperatoren	104
2.6	Kontrollstrukturen	105
2.6.1	Bedingte Verzweigungen	105
2.6.2	Schleifenanweisungen	107
2.6.3	GoTo und GoSub	109
2.7	Zeichenkettenfunktionen	110
2.7.1	Stringverarbeitung	110
2.7.2	Format-Funktion	112
2.8	Vordefinierte Funktionen	116
2.8.1	Mathematische Funktionen	116
2.8.2	Finanzmathematische Funktionen	119

2.8.3	Datums-/Zeitfunktionen	120
2.9	Benutzerdefinierte Funktionen/Prozeduren	123
2.9.1	Funktion	123
2.9.2	Prozedur	124
2.9.3	Parameterübergabe ByRef oder ByVal	124
2.9.4	Optionale Argumente	125
2.9.5	Benannte Argumente	126
2.9.6	Parameter-Arrays	126
2.9.7	Dynamische Arrays als Argumente	127
2.9.8	Rückgabe von Arrays	127
2.9.9	Private-, Public- und Static-Deklarationen	128
2.10	Fehlersuche	130
2.10.1	Direktfenster	131
2.10.2	Verwendung des Debug-Objekts	131
2.10.3	Arbeiten mit dem Lokal-Fenster	132
2.10.4	Überwachungs-Fenster	134
2.10.5	Noch mehr Debugging	135
2.11	Fehlerbehandlung	139
2.11.1	Anweisungen zum Error-Handling	139
2.11.2	Beispiele zum Error-Handling	140
2.11.3	Fehlerbehandlung per Ereignis	142
2.11.4	Fehlerbehandlung komplett deaktivieren	143
2.12	Standarddialogfelder	143
2.12.1	Einfache MsgBox-Anweisung	144
2.12.2	Ausführliche MsgBox-Anweisung	145
2.12.3	Rückgabewerte der MsgBox-Funktion	145
2.12.4	Abfrage von Werten mit der InputBox-Funktion	147
2.13	Übersichten und Ergänzungen	148
2.13.1	Datumskonstanten	148
2.13.2	Rückgabewerte der VarType-Funktion	148
2.14	Praxisbeispiele	149
2.14.1	In einem Textfeld suchen	149
2.14.2	Zeitangaben runden	150
2.14.3	Das Wochenende feststellen	152
2.14.4	Mit dynamischen Arrays rechnen	153
2.14.5	Arbeiten mit dem Debugger	157
3	Makros – eine Einführung	163
3.1	Klassische Makros	163
3.1.1	Entwurfsoberfläche	163

3.1.2	Eigenständige Makros	164
3.1.3	Eingebettete Makros	167
3.1.4	Das AutoKeys-Makro	171
3.1.5	Das AutoExec-Makro	173
3.1.6	Potenziell gefährliche Makroaktionen	173
3.2	Datenmakros	174
3.2.1	Einsatzmöglichkeiten	175
3.2.2	Funktionsprinzip	175
3.2.3	Erzeugen von Datenmakros	176
3.2.4	Datenmakros umbenennen, löschen und ändern	177
3.2.5	USysApplicationLog	177
3.2.6	Aktionen in Datenmakros	178
3.2.7	Auswahl des richtigen Tabellenereignisses	179
3.3	Praxisbeispiele	180
3.3.1	Eingabe-Formular mit neuem Datensatz öffnen	181
3.3.2	Einen Datensatznavigator selbst bauen	182
3.3.3	Ein ereignisgesteuertes Datenmakro erstellen	184
3.3.4	Arbeiten mit einem benannten Datenmakro	189
3.3.5	Per VBA auf ein benanntes Datenmakro zugreifen	193
3.3.6	Änderungen von Tabelleninhalten protokollieren	194
4	Formulare und Steuerelemente	197
4.1	Allgemeines	197
4.1.1	Gruppen von Eigenschaften	198
4.1.2	Methoden	198
4.1.3	Gruppen von Ereignissen	198
4.2	Das Form-Objekt	199
4.2.1	Format-Eigenschaften	199
4.2.2	Daten-Eigenschaften	206
4.2.3	Weitere Eigenschaften	206
4.2.4	Fenster- und Fokus-Ereignisse	208
4.2.5	Tastatur- und Maus-Ereignisse	210
4.2.6	Daten- und Filter-Ereignisse	212
4.2.7	Weitere Ereignisse	214
4.2.8	Methoden	214
4.2.9	Unterformulare	217
4.3	Steuerelemente (Controls)	218
4.3.1	Allgemeines	218
4.3.2	Allgemeine Eigenschaften auf einen Blick	219
4.3.3	Allgemeine Ereignisse auf einen Blick	229

4.3.4	Methoden von Steuerelementen	230
4.3.5	Das Screen-Objekt	231
4.4	ActiveX-Steuerelemente	233
4.4.1	Vergleich mit den integrierten Steuerelementen	233
4.4.2	StatusBar als Beispiel	235
4.5	Praxisbeispiele	239
4.5.1	Das Textfeld programmieren	239
4.5.2	In ungebundene Textfelder ein- und ausgeben	241
4.5.3	Ein ungebundenes Kombinationsfeld füllen	242
4.5.4	Ein Unterformular programmieren	245
4.5.5	Das Register-Steuerelement kennen lernen	248
4.5.6	Die Statusleiste programmieren	252
4.5.7	Verwenden von Bild-Ressourcen	255
4.5.8	Programmieren des Navigationssteuerelements	257
5	Berichte	261
5.1	Allgemeines	261
5.1.1	Reportansichten	261
5.1.2	Die OpenReport-Methode	262
5.1.3	Parameterübergabe	263
5.2	Wichtige Berichtseigenschaften	263
5.2.1	Formateigenschaften	263
5.2.2	Dateneigenschaften	264
5.2.3	Grafikeigenschaften	264
5.2.4	Linien- und Stifteigenschaften	268
5.2.5	Schrifteigenschaften	269
5.2.6	Farb- und Mustereigenschaften	269
5.2.7	Sonstige Eigenschaften	271
5.3	Berichtsergebnisse	273
5.3.1	Allgemeine Ereignisse	273
5.3.2	Tastatur- und Mausereignisse	275
5.4	Berichtsmethoden	276
5.4.1	Grafikmethoden (Übersicht)	276
5.4.2	Scale	276
5.4.3	Line	277
5.4.4	PSet	278
5.4.5	Circle	279
5.4.6	Print	280
5.4.7	TextWidth und TextHeight	281
5.4.8	Sonstige Methoden	282

5.5	Weitere Features des Report-Objekts	282
5.5.1	Rich-Text-Felder drucken	282
5.5.2	Verlauf eines Memofeldes drucken	283
5.5.3	Eine Liste der Anlagen drucken	283
5.5.4	Berichte nachträglich filtern	286
5.5.5	Berichte als PDF-Datei exportieren	287
5.5.6	Berichte als RTF-Datei exportieren	288
5.6	Das Printer-Objekt	288
5.6.1	Wo finde ich das Printer-Objekt?	289
5.6.2	Die Printers-Collection	289
5.6.3	Auswahl eines Druckers	290
5.6.4	Speichern von Berichts-Optionen	292
5.6.5	Eigenschaften des Printers	293
5.7	Direkte Druckausgabe	294
5.8	Übersichten	294
5.8.1	DrawMode-Eigenschaft	294
5.8.2	Farbkonstanten	295
5.9	Praxisbeispiele	295
5.9.1	Aufruf eines Berichts mit Datenfilter	295
5.9.2	Im Report gruppieren und rechnen	299
5.9.3	Erstellen und Drucken eines Diagramms	303
5.9.4	Berichte in Formularen anzeigen	307
6	Programmieren mit Objekten	309
6.1	Objektvariablen	309
6.1.1	Objekttypen und Set-Anweisung	309
6.1.2	Object-Datentyp	311
6.1.3	Form- und Report-Objekt	312
6.1.4	Control-Objekt	313
6.2	Formular- und Berichtsmodule	317
6.2.1	Instanzen von Formularen und Berichten	317
6.2.2	Benutzerdefinierte Form-/Report-Objekte	319
6.2.3	Eigenständige Klassenmodule	320
6.3	Auflistungen	324
6.3.1	Forms/Reports	324
6.3.2	Controls	325
6.3.3	Collection-Objekt	327
6.3.4	Dictionary-Objekt	329
6.3.5	Property und Properties	329
6.3.6	Module-Objekt und Modules-Auflistung	330

6.3.7	Reference-Objekt und References-Auflistung	332
6.4	Die Access-Objekthierarchie	334
6.4.1	Der Objektkatalog	334
6.4.2	Das Application-Objekt allgemein	335
6.4.3	Eigenschaften und Methoden des Application-Objekts	338
6.4.4	Weitere wichtige Objekte	343
6.4.5	AccessObject	344
6.4.6	CurrentProject	345
6.4.7	CurrentData	347
6.5	Übersichten	347
6.5.1	Konstanten der ControlType-Eigenschaft	347
6.5.2	Rückgabewerte der CurrentObjectType-Funktion	348
6.6	Praxisbeispiele	348
6.6.1	Ein Steuerelemente-Array automatisch erstellen	348
6.6.2	Mit Formular-Instanzen arbeiten	352
6.6.3	Mit einer eigenständigen Klasse experimentieren	354
6.6.4	Auf Objekte in Auflistungen zugreifen	357
6.6.5	Properties-Auflistungen untersuchen	360

Teil II: Datenschnittstellen

7	DAO-Programmierung	365
7.1	Allgemeines	365
7.1.1	DBEngine	365
7.1.2	Workspace-Objekt	366
7.1.3	Database-Objekt	367
7.1.4	Recordset-Objekt	367
7.1.5	Verwendung der Datenbankobjekte	368
7.2	Grundlegende Arbeitstechniken	368
7.2.1	Arbeitsumgebung festlegen	369
7.2.2	Datenbank anlegen und öffnen	369
7.2.3	Tabellen/Indizes anlegen	373
7.2.4	Tabellen einbinden	378
7.2.5	Tabellen verknüpfen (Relationen)	379
7.2.6	Abfragen erstellen/ausführen	381
7.2.7	Öffnen von Tabellen/Abfragen	383
7.3	Arbeiten mit Recordsets	386
7.3.1	Eigenschaften und Methoden von Recordsets	386
7.3.2	Datensätze anzeigen	389
7.3.3	Datensätze hinzufügen/ändern	391

7.3.4	Datensätze löschen	393
7.3.5	Datensätze sortieren	395
7.3.6	Datensätze suchen	396
7.3.7	Datensätze filtern	397
7.3.8	DAO in gebundenen Formularen	398
7.3.9	Auf Anlage-Felder zugreifen	401
7.3.10	Auf mehrwertige Felder zugreifen	404
7.3.11	Verlaufsverfolgung eines Memo-Felds	405
7.4	Weitere Funktionen	406
7.4.1	Eigenschaften (Properties)	406
7.4.2	Transaktionen	408
7.5	Praxisbeispiele	409
7.5.1	Eine Tabelle anlegen	409
7.5.2	Navigieren mit DAO	412
7.5.3	Den Datensatzzeiger bewegen	415
7.5.4	In Recordsets suchen	419
7.5.5	Eine Datenbank analysieren	422
7.6	Komplexbeispiel: Telefonverzeichnis	425
7.6.1	Eingabemaske	425
7.6.2	Anforderungen	425
7.6.3	Programmierung	426
7.6.4	Test und Bemerkungen	435
8	ADO-Programmierung	437
8.1	Ein erster Blick auf ADO	437
8.1.1	Kleines Einführungsbeispiel	438
8.1.2	Zur Geschichte von ADO	439
8.1.3	Hinweise zu den ADO-Bibliotheken	440
8.1.4	ADO und OLE DB	441
8.1.1	ADO-Objektmodell	442
8.2	ADO-Grundoperationen	444
8.2.1	Beziehungen zwischen den Objekten	444
8.2.2	Die Verbindung zur Datenquelle	445
8.2.3	Aktionsabfragen mit dem Command-Objekt	449
8.2.4	Recordsets mit Daten füllen	451
8.3	Weitere Operationen mit Recordsets	456
8.3.1	Welche Recordset-Features werden unterstützt?	456
8.3.2	Editieren von Datensätzen	457
8.3.3	Hinzufügen von Datensätzen	458
8.3.4	Löschen von Datensätzen	458

8.3.5	Recordsets filtern	459
8.3.6	Ungebundene Recordsets	460
8.3.7	Recordsets abspeichern	461
8.3.8	Bewegen in Recordsets	461
8.3.9	Daten direkt einlesen	462
8.3.10	Sortieren	463
8.3.11	Suchen	464
8.3.12	Ereignisse auswerten	464
8.4	Zugriff auf ADO-Auflistungen	466
8.4.1	Allgemeine Features	466
8.4.2	Property und Properties	467
8.4.3	Field und Fields	468
8.4.4	Parameter und Parameters	469
8.4.5	Error und Errors	470
8.5	Übersichten	471
8.5.1	Connection-Objekt	471
8.5.2	Command-Objekt	472
8.5.3	Recordset-Objekt	472
8.6	Praxisbeispiele	474
8.6.1	Mit ADO auf eine Access-Datenbank zugreifen	474
8.6.2	Ein ADO-Datenklassenmodul verwenden	476
8.6.3	Ein intelligentes ADO-Frontend entwickeln	479
9	Datenbankverwaltung	485
9.1	Datenbankverwaltung mit ADOX	485
9.1.1	Datenbanken erstellen	487
9.1.2	Tabellendefinition	489
9.1.3	Indexdefinition	493
9.1.4	Erstellen von Prozeduren und Sichten	495
9.1.5	Tabellen verknüpfen (Relationen)	496
9.2	Erstellen spezieller Feldtypen	497
9.2.1	Automatische Zufallswerte (GUID)	497
9.2.2	Memofeld mit Archiv-Funktion (Nur anfügen)	499
9.2.3	Anlage-Feld	501
9.2.4	Rich-Text-Feld	502
9.2.5	Multivalue-Feld (MVF)	503
9.2.6	Berechnete Spalten	508
9.2.7	Beschreibung von Datenbankfeldern setzen	510
9.3	Zugriffsschutz in Access-Datenbanken	512
9.3.1	Grundlagen	512

9.3.2	Sichern auf Datenbankebene (DAO)	514
9.3.3	Sichern auf Datenbankebene (ADO/ADOX)	515
9.3.4	Erstellen neuer Benutzer und Gruppen (DAO)	515
9.3.5	Vergabe von Rechten (DAO)	517
9.3.6	Komplettbeispiel: Nutzerbasierte Sicherheit	519
9.3.7	Erstellen neuer Benutzer und Gruppen (ADOX)	524
9.3.8	Vergabe von Rechten (ADOX)	525
9.3.9	Verschlüsseln von Datenbanken	527
9.4	Multiuserzugriff	530
9.4.1	Verwenden der DAO	531
9.4.2	Verwenden der ADO	534
9.5	ODBC-Verbindungen	535
9.5.1	Ein Blick auf den ODBC-Datenquellen-Administrator	535
9.5.2	Erstellen einer ODBC-Verbindung (DAO)	537
9.5.3	Öffnen einer ODBC-Verbindung (DAO)	538
9.5.4	Öffnen einer ODBC-Verbindung (ADO)	540
9.5.5	Konfigurieren von ODBC-Verbindungen	541
9.6	Zugriff auf Fremdformate	541
9.6.1	dBASE III/IV- und FoxPro-Datenbanken	542
9.6.2	Textdateien (TXT/ASC/CSV)	546
9.7	Einbinden externer Tabellen	550
9.7.1	Verwenden der DAO	550
9.7.2	Verwenden der ADOX	552
9.8	Exportieren von Daten	554
9.8.1	TransferDatabase-Methode	554
9.8.2	Exportieren mit SQL-Anweisungen	555
9.9	Replizieren von Datenbanken	555
9.10	Optimierung	556
9.10.1	Indizes	556
9.10.2	Abfrage-Optimierung	556
9.10.3	Weitere Möglichkeiten	557
9.10.4	ADO/DAO/ODBC – Was ist schneller?	558
9.11	Tipps & Tricks	560
9.11.1	Wie prüft man die ADO-Versionsnummer?	560
9.11.2	Access-Datenbanken exklusiv öffnen	560
9.11.3	Access-Datenbanken im Netzwerk	561
9.11.4	Alle aktiven Verbindungen zur Datenbank auflisten	561
9.11.5	Das Datenbank-Kennwort ändern	562
9.11.6	Abfragen über mehrere Datenbanken	563
9.11.7	Datenbanken reparieren/komprimieren	563

10	Microsoft SQL Server	565
10.1	Ein erster Schock ...	565
10.2	Allgemeines	566
10.2.1	SQL Server LocalDB	567
10.2.2	SQL Server Express	568
10.2.3	Unterschiede SQL Server-Varianten/Jet-Engine	569
10.2.4	Client- versus Fileserver-Programmierung	571
10.2.5	Installation SQL Server Express	573
10.2.6	Netzwerkzugriff für den SQL Server Express	577
10.2.7	Die wichtigsten Tools von SQL Server	579
10.2.8	Vordefinierte Datenbanken	582
10.2.9	Einschränkungen	583
10.2.10	Weitere SQL Server-Funktionen im Kurzüberblick	584
10.2.11	Datenbanken verwalten	585
10.3	Transact-SQL – die Sprache des SQL Servers	587
10.3.1	Schreibweise	587
10.3.2	Kommentare	588
10.3.3	Zeichenketten	588
10.3.4	Variablen deklarieren/verwenden	589
10.3.5	Bedingungen mit IF/ELSE auswerten	590
10.3.6	Verwenden von CASE	591
10.3.7	Verwenden von WHILE...BREAK/CONTINUE	591
10.3.8	Datum und Uhrzeit in T-SQL	592
10.3.9	Verwenden von GOTO	592
10.4	Praktisches Arbeiten mit dem SQL Server	593
10.4.1	Erstellen neuer SQL Server-Datenbanken	593
10.4.2	Erzeugen und Verwalten von Tabellen	594
10.4.3	Erzeugen und Verwenden von Sichten (Views)	595
10.4.4	Verwenden von Gespeicherten Prozeduren	597
10.4.5	Programmieren von Triggern	601
10.4.6	Erzeugen von Datenbankdiagrammen	605
10.4.7	Volltextabfragen	606
10.4.8	Datenbanken sichern und wiederherstellen	612
10.5	Fehlerbehandlung	615
10.5.1	Das Fehlermodell des SQL Servers	616
10.5.2	Verwenden von @@ERROR	616
10.5.3	Verwenden von RAISEERROR	617
10.5.4	Fehlerbehandlung mit TRY...CATCH	618
10.5.5	Fehlerbehandlung mit den ADO	620
10.6	Datensicherheit auf dem Microsoft SQL Server	622

10.6.1	Überblick Sicherheitsmodell	623
10.6.2	Verwalten mit dem SQL Server Management Studio	625
10.6.3	Verwalten mit T-SQL	629
10.7	Tipps & Tricks	632
10.7.1	Alle registrierten Microsoft SQL Server ermitteln	632
10.7.2	Alle Datenbanken ermitteln	633
10.7.3	Alle Tabellen ermitteln	633
10.7.4	Eine Tabelle löschen	634
10.7.5	Anzahl der Datensätze beschränken	635
10.7.6	Platzhalterzeichen in TSQL	636
10.7.7	Leerzeichen entfernen	636
10.7.8	Teilstrings erzeugen	636
10.7.9	Mit einer Datenbankdatei verbinden	637
10.7.10	Warum wird @@ERROR nicht korrekt verarbeitet?	638
10.7.11	Die Anzahl der Datensätze bestimmen	638
10.7.12	Warum sind Abfragen mit Platzhaltern so langsam?	638
10.7.13	Groß-/Kleinschreibung berücksichtigen	639
10.7.14	Das Ergebnis einer Stored Procedure speichern	639
10.7.15	Eine Datenbank umbenennen	639
10.7.16	Eine Datenbank zwischen Servern verschieben	640
10.7.17	Die Datenbankstruktur kopieren	641
10.7.18	Nach dem Löschen IDENTITY auf 0 setzen	642
10.7.19	Eine Tabellenspalte umbenennen	642
10.7.20	Temporäre Tabellen unterscheiden	642
10.7.21	Daten aus verschiedenen Datenbanken anzeigen	643
10.7.22	Einen SMO-Mapper realisieren	643
10.8	Übersichten	648
10.8.1	Datentypen	648
10.8.2	Unterschiede Access- und SQL Server-Datentypen	649
11	Access und Azure SQL	651
11.1	Einführung in SQL Azure-Datenbanken	652
11.1.1	Das Grundprinzip der "Webdatenbank"	652
11.1.2	Der Azure-Server	654
11.1.3	Die Frage nach den Kosten	655
11.2	Einrichten des Servers	656
11.2.1	Die zentrale Organisationsstruktur	657
11.2.2	Einen Server und eine Datenbank erstellen	658
11.2.3	IP-Filter konfigurieren	662
11.2.4	Bemerkungen zum neu erstellten Account	663

11.2.5	Die drei konzeptionellen Zugriffsmodelle	663
11.3	Administrieren von Azure SQL-Datenbanken	666
11.3.1	Zugriff mit dem SQL Server Management Studio	666
11.3.2	Weitere Accounts erstellen	668
11.3.3	Lokale Datenbanken migrieren	671
11.3.4	Migrieren von Access-Datenbanken	674
11.4	Praktische Umsetzung in Access	676
11.4.1	Tabellen einbinden	677
11.4.2	DAO- oder ADO-Zugriff – keine Frage!	680
11.4.3	Unsere AzureSQL-Library	680
11.4.4	Verbindung mit ADO aufbauen	681
11.4.5	Datenbank erstellen	685
11.4.6	Ist die Datenbank schon vorhanden?	686
11.4.7	Den aktuellen "Füllstand" abrufen	687
11.4.8	Was passiert, wenn die Datenbank zu klein wird?	690
11.4.9	Eine Datenbankkopie erstellen	690
11.4.10	Tabelle(n) erstellen	691
11.4.11	Daten exportieren	692
11.4.12	Daten einbinden	693
11.4.13	Daten lesen	694
11.4.14	Daten schreiben	695
11.5	Abschließende Hinweise	696
11.5.1	Synchronisieren	696
11.5.2	Performance-Tipps	697
11.5.3	Die Firewall per T-SQL konfigurieren	697
11.5.4	Arbeiten mit sqlcmd	699
11.6	Fazit	699
12	Zugriff auf SQLite	701
12.1	Was eigentlich ist SQLite?	702
12.1.1	Vorteile	702
12.1.2	Nachteile	703
12.2	Vorbereitungen	703
12.2.1	Download/Installation des ODBC-Treibers	704
12.2.2	Download/Installation SQLite for Excel	705
12.3	Datenbank-Tools	706
12.3.1	Database .NET	706
12.3.2	SQLite Administrator	708
12.4	Praktische Aufgabenstellungen	709
12.4.1	Einbinden von SQLite-Tabellen per Assistent	709

12.4.2	Einbinden von SQLite-Tabellen per VBA-Code	710
12.4.3	Datenbank per Code erstellen	711
12.4.4	Tabellen erzeugen	712
12.4.5	Datenbankzugriff per ADO realisieren	714
12.4.6	Die Bedeutung von Transaktionen bei SQLite	715
12.4.7	SOUNDEX verwenden	716
12.4.8	Volltextabfragen realisieren	717
12.5	Praxisbeispiele	720
12.5.1	Verwenden der Volltextsuche	720
12.5.2	Implementieren der Klasse SQLiteDatabase	725
12.5.3	Verwenden der Klasse SQLiteDatabase	735
12.6	Tipps & Tricks	737
12.6.1	Für Liebhaber der Kommandozeile – Sqlite3.exe	737
12.6.2	Eine SQLite-Datenbank reparieren	739
12.6.3	Eine Beispieldatenbank herunterladen	740
12.6.4	Testen ob Tabelle vorhanden ist	740
12.6.5	Die Datenbank defragmentieren	740
12.6.6	Mehrere Datenbanken verknüpfen	741
12.6.7	Eine Abfrage/Tabelle kopieren	741
12.6.8	Ein Backup implementieren	742
12.6.9	Tabellen zwischen Datenbanken kopieren	742
12.6.10	Ersatz für TOP	742
12.6.11	Metadaten auswerten	743
12.6.12	Timestamp als Defaultwert verwenden	744
12.6.13	Export in XML-Format	745
12.7	Fazit	745

Teil III: Weitere Technologien

13	Dateien und Verzeichnisse	749
13.1	Allgemeines	749
13.1.1	ANSI/ASCII/Unicode	749
13.1.2	Gemeinsamer Dateizugriff	750
13.1.3	Verwenden der File System Objects	750
13.2	Zugriff auf das Dateisystem	752
13.2.1	Ermitteln aller Laufwerke und deren Eigenschaften	752
13.2.2	Ermitteln aller Verzeichnis-Eigenschaften	753
13.2.3	Auflisten aller Unterverzeichnisse eines Folders	754
13.2.4	Rekursiv alle Unterverzeichnisse auflisten	755
13.2.5	Ein Verzeichnis erzeugen	756

13.2.6	Das Datenbankverzeichnis bestimmen	757
13.2.7	Abfragen des Temp-/System/...-Verzeichnisses	757
13.2.8	Prüfen, ob eine Datei existiert	758
13.2.9	Verzeichnisse/Dateien kopieren/löschen	759
13.2.10	Auflisten aller Dateien eines Verzeichnisses	759
13.2.11	Name, Pfad und Extension einer Datei ermitteln	760
13.2.12	Einen Tempfile-Namen erzeugen	762
13.3	Textdateien	762
13.3.1	Klassischer Zugriff auf Textdateien	763
13.3.2	Zugriff auf Textdateien mit den File System Objects	765
13.4	Typisierte Dateien	768
13.4.1	Öffnen	768
13.4.2	Lesen/Schreiben	768
13.5	Weitere Dateien	770
13.5.1	Binärdateien	770
13.5.2	INI-Dateien	771
13.6	Die Registrierdatenbank	773
13.6.1	Einführung	773
13.6.2	API/VBA-Zugriff auf die Registrierungsdatenbank	774
13.6.3	API-Konstanten/Funktionen für den Registry-Zugriff	775
13.6.4	Prüfen, ob ein Schlüssel existiert	777
13.6.5	Einen vorhandenen Wert auslesen	777
13.6.6	Einen Schlüssel erstellen	778
13.6.7	Einen Wert setzen bzw. ändern	779
13.6.8	Einen Schlüssel löschen	779
13.6.9	Ein Feld löschen	779
13.6.10	Aufruf der Funktionen	780
13.7	Dateidialoge	782
13.7.1	Variante 1 (Office 16 Library)	783
13.7.2	Variante 2 (Windows-API)	785
13.7.3	Verzeichnisdialog (Windows-API)	790
13.8	Übersichten	792
13.8.1	Dateifunktionen in Access	792
13.8.2	FSO-Eigenschaften und -Methoden	793
13.9	Praxisbeispiele	794
13.9.1	Auf eine Textdatei zugreifen	794
13.9.2	Dateien suchen	799
13.9.3	Die Shellfunktionen verwenden	801
13.9.4	Einen Verzeichnisbaum in eine TreeView einlesen	806

14	XML in Theorie und Praxis	807
14.1	XML – etwas Theorie	807
14.1.1	Allgemeines	807
14.1.2	Der XML-Grundaufbau	810
14.1.3	Wohlgeformte Dokumente	811
14.1.4	Processing Instructions (PI)	813
14.1.5	Elemente und Attribute	814
14.1.6	Verwendbare Zeichensätze	815
14.2	XSD-Schemas	817
14.2.1	Das Grundprinzip	817
14.2.2	Ein XSD-Schema mit Microsoft Access erzeugen	819
14.3	XML-Verarbeitung mit dem DOM	823
14.3.1	Was ist das DOM?	823
14.3.2	Erste Schritte	824
14.3.3	Erzeugen von Instanzen	825
14.3.4	Laden von Dokumenten	826
14.3.5	XML-Fehlerprüfung/-Analyse	827
14.3.6	Erzeugen von XML-Dokumenten	829
14.3.7	Auslesen von XML-Dateien	831
14.3.8	Einfügen von Informationen	834
14.3.9	Attribute oder Element	837
14.3.10	Suchen in den Baumzweigen	840
14.3.11	Das Interpretieren von Leerzeichen	843
14.4	XML-Integration in Access	846
14.4.1	Importieren	846
14.4.2	Exportieren	849
14.4.3	XML-Transformation mit XSLT	853
14.4.4	Application-Objekt (ImportXML/ExportXML)	856
14.4.5	ADO-Recordset	859
14.5	Vor- und Nachteile von XML	861
14.5.1	Grundsätzlicher Vergleich	862
14.5.2	Zeitvergleich ADO/XML	862
14.6	Praxisbeispiele	864
14.6.1	Speichern im UTF-8-/UTF-16-Format	864
14.6.2	UTF-8-/UTF-16 aus einem ADO-Stream laden	866
14.6.3	XML-Daten asynchron verarbeiten	868
14.6.4	XML-Daten in einer TreeView darstellen	870
14.6.5	Navigieren zwischen einzelnen XML-Baumknoten	873
14.6.6	ADO-XML-Streams nachbearbeiten	876
14.6.7	Textdaten in XML-Dokumente umwandeln	879

15	SQL im Einsatz	881
15.1	Einführung	881
15.1.1	SQL-Dialekte	882
15.1.2	Kategorien von SQL-Anweisungen	882
15.2	Etwas (Datenbank-)Theorie	884
15.2.1	Allgemeines/Begriffe	884
15.2.2	Normalisieren von Tabellen	889
15.2.3	Beziehungen zwischen den Tabellen	894
15.2.4	Verknüpfen von Tabellen	896
15.3	Testprogramm und Beispieldatenbank	900
15.3.1	Hinweise zur Bedienung	900
15.3.2	Die Beispieldatenbank im Überblick	900
15.3.3	ADO Query	901
15.3.4	Bemerkungen	902
15.4	Daten abfragen	903
15.4.1	Abfragen mit SELECT	904
15.4.2	Alle Spalten auswählen	904
15.4.3	Auswahl der Spalten	905
15.4.4	Filtern	906
15.4.5	Beschränken der Ergebnismenge	912
15.4.6	Eindeutige Records/doppelte Datensätze	913
15.4.7	Tabellen verknüpfen	914
15.4.8	Tabellen vereinigen	917
15.4.9	Datensätze sortieren	918
15.4.10	Datensätze gruppieren	918
15.4.11	Unterabfragen	920
15.4.12	Anlage-Felder mit SQL verwalten	924
15.4.13	History-Felder mit SQL abfragen	926
15.4.14	Mehrwertige Felder mit SQL abfragen	928
15.5	Daten manipulieren	929
15.5.1	Einfügen einzelner Datensätze	930
15.5.2	Einfügen von Abfragedaten	931
15.5.3	Exportieren/Importieren von Abfragedaten	933
15.5.4	Aktualisieren/Ändern	935
15.5.5	Löschen	936
15.6	Erweiterte SQL-Funktionen	937
15.6.1	Berechnete/Formatierte Spalten	938
15.6.2	Berechnungsfunktionen	944
15.6.3	NULL-Werte	945
15.6.4	Datum und Zeit in SQL-Abfragen	947

15.6.5	Datentypumwandlungen	952
15.6.6	Kreuztabellenabfragen	953
15.6.7	Steuerelemente in Abfragen verwenden	956
15.6.8	Globale Variablen in Abfragen verwenden	957
15.7	Datenbankverwaltung mit SQL (DDL)	957
15.7.1	Datenbanken	958
15.7.2	Tabellen	958
15.7.3	Indizes	959
15.7.4	Tabellen/Indizes löschen oder verändern	962
15.7.5	Sichten (Views)	962
15.7.6	Nutzer- und Rechteverwaltung	964
15.7.7	Transaktionen	964
15.8	SQL in der Access-Praxis	965
15.8.1	SQL in Abfragen	965
15.8.2	SQL im Basic-Code	966
15.8.3	SQL beim Oberflächenentwurf	968
15.8.4	VBA-Funktionen in SQL-Anweisungen	968
15.9	Tipps & Tricks	970
15.9.1	Wie kann ich die Anzahl der Datensätze ermitteln?	970
15.9.2	Wie nutze ich Datumsteile in SQL zur Suche?	971
15.9.3	Die Groß-/Kleinschreibung berücksichtigen	971
15.9.4	Warum erhalte ich zu viele Datensätze ?	972
15.9.5	Doppelte Datensätze aus einer Tabelle löschen	973
15.10	Praxisbeispiele	974
15.10.1	Ein komfortables SQL-Abfrageprogramm erstellen	974
15.10.2	Datum und Zeit in SQL einbauen	976
16	Anwendungsdesign	979
16.1	Access-Oberflächengestaltung	979
16.1.1	Beschriften der Kopfzeile	980
16.1.2	Informationen in der Statuszeile anzeigen	982
16.1.3	Fortschrittsanzeige mit dem Progressbar realisieren	983
16.1.4	Navigationsbereich konfigurieren	984
16.1.5	Access-Hauptfenster komplett ausblenden	990
16.2	Steuern der Anwendung	993
16.2.1	Autostart mit AutoExec-Makro	993
16.2.2	Formulare automatisch starten	995
16.2.3	Warten auf das Ende	995
16.2.4	Access per VBA beenden	996
16.2.5	Den Runtime-Modus aktivieren	996

16.2.6	Befehlszeilen-Optionen verwenden	997
16.3	Entwickeln von Assistenten und Add-Ins	999
16.3.1	Assistenten-Typen	999
16.3.2	Einbinden der Assistenten in die Access-IDE	999
16.3.3	Menü-Assistent (Beispiel)	1001
16.3.4	Objekt-Assistent (Beispiel)	1003
16.3.5	Steuerelemente-Assistent (Beispiel)	1007
16.3.6	Eigenschaften-Assistent	1010
16.4	Entwickeln/Einbinden von Managed Add-Ins	1011
16.4.1	Interaktion Anwendung/Add-In	1011
16.4.2	Entwurf des Add-Ins	1012
16.4.3	Oberfläche	1014
16.4.4	Der Quellcode des Add-Ins	1015
16.4.5	Formularentwurf für das Add-In	1017
16.4.6	Kompilieren und Einbinden	1017
16.4.7	Testen	1018
16.5	Libraries unter Access	1019
16.5.1	Erstellen und Einbinden	1019
16.5.2	Debugging	1020
16.5.3	Einfaches Beispiel	1021
16.5.4	Hinweise	1023
16.6	Praxisbeispiele	1024
16.6.1	Mehr über die aktuelle Access-Version erfahren	1024
16.6.2	Access neu starten/Datenbank neu laden	1025
16.6.3	Mit Reference-Objekten arbeiten	1026
16.6.4	Benutzerdefinierte Eigenschaften einsetzen	1030
16.6.5	Den aktuellen Datenbanknutzer ermitteln	1034
16.6.6	Überlappende Fenster einstellen	1034
16.6.7	Access-Optionen abrufen/setzen	1035
17	Menüband und Backstage	1037
17.1	Allgemeine Grundlagen	1037
17.1.1	Manuelle Konfigurationsmöglichkeiten	1038
17.1.2	Grundprinzip der Programmierung	1039
17.1.3	Verwenden der Tabelle USysRibbons	1041
17.1.4	Application.LoadCustomUI als Alternative	1042
17.2	Ein kleines Testprogramm	1042
17.3	Praktische Aufgabenstellungen	1044
17.3.1	Infos über Steuerelemente und Symbole erhalten	1044
17.3.2	Hauptregisterkarten ausblenden	1046

17.3.3	Einzelne Registerkarten ausblenden	1047
17.3.4	Einzelne Gruppen ausblenden	1048
17.3.5	Registerkarten, Gruppen und Schaltflächen einfügen	1049
17.3.6	Ereignisbehandlung mit VBA-Code/Makros	1051
17.3.7	Verändern von Eigenschaften mit VBA-Callbacks	1052
17.3.8	Aktualisieren des Menübands per VBA-Code	1052
17.3.9	Kontextabhängige Registerkarten	1053
17.3.10	Registerkarten per VBA aktivieren	1054
17.3.11	Fehlermeldungen des Menübands anzeigen	1055
17.3.12	Vorhandene Funktionen des Menübands ändern	1055
17.3.13	Die Schnellzugriffsleiste erweitern/programmieren	1056
17.4	Übersicht der Steuerelemente	1057
17.4.1	labelControl-Steuerelement	1057
17.4.2	button-Steuerelement	1057
17.4.3	separator-Steuerelement	1061
17.4.4	toggleButton-Steuerelement	1062
17.4.5	buttonGroup-Steuerelement	1063
17.4.6	checkBox-Steuerelement	1064
17.4.7	editBox-Steuerelement	1065
17.4.8	comboBox-Steuerelement	1066
17.4.9	dropDownElement-Steuerelement	1070
17.4.10	gallery-Steuerelement	1073
17.4.11	menu-Steuerelement	1075
17.4.12	splitButton-Steuerelement	1078
17.4.13	dialogBoxLauncher	1079
17.5	Die Backstage-Ansicht anpassen	1081
17.5.1	Die Standardansicht verändern	1082
17.5.2	Die drei möglichen Layouts für Registerkarten	1084
17.5.3	Die neuen Möglichkeiten von Gruppen	1091
17.5.4	Verwenden von LayoutContainern	1093
17.5.5	Das Verhalten der Schaltflächen beeinflussen	1094
17.5.6	Mit VBA/Makros auf Ereignisse reagieren	1095
17.6	Tipps & Tricks	1096
17.6.1	Die guten alten Access 2003-Menüs anzeigen	1096
17.6.2	Das Office-Menü für Access 2007 anpassen	1098
17.6.3	XML-Daten komfortabel editieren	1099
17.6.4	Arbeiten mit dem RibbonCreator	1100
17.7	Übersichten	1101

18	Programmschnittstellen	1103
18.1	Zwischenablage	1103
18.1.1	Kopieren/Einfügen mittels DoCmd-Objekt	1103
18.1.2	Ein Clipboard-Objekt programmieren	1104
18.2	API- und DLL-Einbindung	1107
18.2.1	Allgemeines	1107
18.2.2	Und was ist mit der 64-Bit Access Version?	1108
18.2.3	Woher bekomme ich Infos über die Win32-API?	1110
18.2.4	Einbinden der Deklaration	1111
18.2.5	Wert oder Zeiger?	1113
18.2.6	Übergabe von Strings	1114
18.2.7	Verwenden von As Any	1115
18.2.8	Übergabe von Arrays	1115
18.2.9	Besonderheiten mit Records	1116
18.2.10	Zuordnen der Datentypen	1118
18.3	OLE/ActiveX	1119
18.3.1	Überblick	1119
18.3.2	OLE	1120
18.3.3	Übersicht zum Objektfeld-Steuerelement	1121
18.3.4	Programmieren mit dem Objektfeld	1124
18.3.5	ActiveX-Code-Komponenten (OLE-Automation)	1127
18.3.6	Programmieren mit ActiveX-Code-Komponenten	1128
18.4	DDE	1130
18.4.1	Funktionsprinzip	1130
18.4.2	Verwenden von Steuerelementen	1131
18.4.3	VBA-Programmierung	1132
18.5	Scanner-Unterstützung per WIA	1134
18.5.1	Was kann WIA?	1134
18.5.2	Installation/Vorbereitung	1134
18.5.3	Einstieg mit kleiner Beispielanwendung	1135
18.5.4	Reagieren auf das Verbinden/Trennen von Geräten	1136
18.5.5	Ermitteln der verfügbaren Geräte	1137
18.5.6	Anzeige der Geräteeigenschaften	1137
18.5.7	Ein Bild einlesen	1139
18.5.8	Bild(er) drucken (Assistent)	1140
18.5.9	Den Scanner-Assistent aufrufen	1141
18.5.10	Grafikbearbeitung	1142
18.6	Zugriff auf Excel-Arbeitsmapen	1144
18.6.1	Zugriffsmöglichkeiten	1144
18.6.2	TransferSpreadsheet	1145

18.6.3	Import/Export per DAO	1149
18.6.4	Daten auslesen	1153
18.6.5	Zugriff auf Tabellenbereiche	1155
18.6.6	OLE-Automation	1156
18.7	Praxisbeispiele	1159
18.7.1	Eine externe Anwendung starten	1159
18.7.2	Informationen über das aktuelle System ermitteln	1161
18.7.3	Das aktuelle Betriebssystem ermitteln	1164
18.7.4	Den Windows-Lizenznehmer ermitteln	1167
18.7.5	Zeitmessungen in Access durchführen	1169
18.7.6	Microsoft Word über ActiveX steuern	1173
18.7.7	Microsoft Excel über ActiveX steuern	1177
	Index	1179



Vorwort

Microsoft Access ist weit mehr als nur eine komfortable Datenbankverwaltung, denn dem fortgeschrittenen Anwender steht darüber hinaus eine vollständige und vor allem preiswerte Entwicklungsumgebung für individuell geprägte Datenbankanwendungen zur Verfügung. Zusammen mit der objekt- und ereignisorientierten Programmieretechnik bilden VBA, SQL, DAO/ADO, Makros und XML ein mächtiges Werkzeug, um leistungsfähige Datenbank-Applikationen mit besserer Performance und auf beachtlich höherem Niveau zu erstellen.

Zum Buchinhalt

HINWEIS: Dieses Buch bietet Ihnen nicht nur eine fundierte Einführung in die Programmierung von Datenbanken mit Access 2016, sondern ist gleichermaßen auch für die Versionen 2010 und 2013 geeignet.

Die Autoren haben bei der Zusammenstellung des Inhalts weniger aus den zu Access 2016 mitgelieferten Dokumentationen, sondern vor allem aus eigenen Quellen und langjährigen praktischen Erfahrungen geschöpft:

- Lehrgänge für Datenbankprogrammierer
- Vorlesungen über Datenbankprogrammierung an Fachhochschulen
- und (last, but not least) das zahlreiche Feedback unserer Leser zu unseren Vorgängertiteln

Mit VBA (*Visual Basic for Applications*) stoßen Sie das Tor zur Windows-Programmierung weit auf und greifen direkt auf die mächtige Access-Bibliothek, die DAO-/ADO-Datenzugriffsobjekte oder auf das Windows-API (*Application Programming Interface*) zu. Verständlicherweise ist es im Rahmen dieses Buches unmöglich, auf alle Funktionen sowie auf alle Objekte und Sprachelemente der einzelnen Bibliotheken einzugehen. Dies ist Sache der Befehlsreferenz, auf die Sie am bequemsten über die integrierte Online-Hilfedatei zugreifen können. Ziel des Buches soll es sein, einen Gesamtüberblick zu geben und praktische Konzepte zu vermitteln.

In den insgesamt 18 Kapiteln finden Sie:

- Ausführliches Know-how über die objekt- und ereignisorientierte Gestaltung der Benutzerschnittstelle von Microsoft Access
- Eine gestraffte Einführung in die Programmierung von Makros
- Eine umfassende Einführung in VBA, SQL und XML mit einer übersichtlichen und auf das Wesentliche reduzierten Sprachbeschreibung
- Eine gestraffte Einführung in den Datenbankzugriff (Jet, Microsoft SQL Server, Azure SQL) unter Verwendung von DAO/ADO
- Viele Kapitel verfügen über einen Übersichtsteil, in dem wichtige Informationen (z.B. relevante Eigenschaften, Methoden und Ereignisse von zentralen Objekten) griffbereit zur Verfügung stehen
- Zahlreiche Praxisbeispiele am Ende der Kapitel dienen der Vertiefung der vermittelten theoretischen Grundlagen.

Begleitdateien

Den Quellcode dieses Buchs können Sie sich unter folgender Adresse herunterladen:

LINK: www.doko-buch.de

Für Einsteiger einige Hinweise, die wir aufgrund von Erfahrungen mit unseren Vorgängertiteln diesmal nicht vergessen wollen:

- Sie sollten natürlich vorher Microsoft Access 2016/2013/2010 auf Ihrem PC installiert haben.
- Sollte doch einmal eine Datenbank/Datei beim Testen der Beispiele nicht gefunden werden, müssen Sie im betreffenden Modul die Pfadangaben anpassen.
- Falls einmal ein Beispiel nicht funktionieren sollte, lesen Sie die beigefügte Readme-Datei.
- Zumindest für das Kapitel 10 ist die Installation des Microsoft SQL Server ab Version 2008 bzw. der entsprechenden Express Version erforderlich. Beachten Sie die erforderlichen Sicherheitseinstellungen für den Zugriff.

Sicherheitseinstellungen

Im Zeitalter wachsender Internetkriminalität müssen Sie sich daran gewöhnen, dass zum Schutz Ihres Computers reichlich Sicherheitshürden aufgebaut wurden: Access-Sicherheit (VBA), System-sicherheit (Windows 7/8/10), SQL Server-Sicherheit, .NET-Sicherheit.

Das führt für Sie als VBA-Programmierer zunächst zu folgender Konsequenz:

HINWEIS: Aufgrund der restriktiven Sicherheitsanforderungen von Access 2016 wird es Ihnen in der Regel nicht gelingen, ohne entsprechende Anpassungen der Entwicklungsumgebung, ihren eigenen Code bzw. die zu diesem Buch mitgelieferten Beispiele zum Laufen zu bringen!

Auch unter Access 2016 wird VBA-Code standardmäßig als "unsicherer Code" eingestuft. Falls das Formular den Code nicht ausführt und stattdessen unterhalb des Menübands eine Sicherheitswarnung zeigt, klicken Sie auf die *Inhalt aktivieren*-Schaltfläche.

Um diese lästigen Sicherheitswarnungen generell zu verhindern, sollten Sie (zumindest für die Dauer der Programmentwicklung) im *Sicherheitscenter* Ihr Datenbankverzeichnis als *Vertrauenswürdigen Speicherort* hinzufügen.

HINWEIS: Eine detaillierte Anleitung, wie Sie mit diesem "heißen Eisen" am besten umgehen, finden Sie im Abschnitt "Sicherheitseinstellungen" des Kapitels 1 (Seite 43).

Ein gutgemeinter Ratschlag an Einsteiger

Programmieren lernt man nur durch Beispiele! Dies ist eine knallharte Wahrheit, um die sich eine zu stark akademisch geprägte Leserschaft gern herumogeln möchte. Wie alle unsere anderen Programmier-Bücher ist deshalb auch dieses Buch kein Lehrbuch, das Sie nach dem Motto "Jetzt lernen wir Microsoft Access" linear von vorn nach hinten durcharbeiten sollen. Sehr schnell wird Ihnen dabei unterwegs die Luft ausgehen und Sie verlieren die Lust. Falls Sie es dennoch bis zum Ende schaffen, sind Sie dennoch nicht in der Lage, eine praxistaugliche Datenbankapplikation zu erstellen.

Wir empfehlen hingegen dem Einsteiger, sich zunächst nur auf einige ausgewählte Kapitel zu konzentrieren und parallel zum Studium des Buchs und tatkräftiger Einbeziehung der Online-Hilfe ein privates Datenbankprojekt in Angriff zu nehmen, wie zum Beispiel die Verwaltung der eigenen Bücher- oder CD-Sammlung. Der Lerneffekt, der aus selbst begangenen und selbst behobenen Fehlern resultiert, ist ein ungeheurer!

Motiviert durch kleine Erfolgserlebnisse und Aha-Effekte werden Sie Ihr Projekt und damit Ihr Wissen schrittweise vergrößern und verfeinern und sich nach dem Prinzip "soviel wie nötig" nur auf die Kapitel konzentrieren, die Sie zur Lösung Ihrer aktuellen Probleme wirklich benötigen.

Nobody is perfect

Sie werden – trotz der 1200 Seiten – in diesem Buch nicht alles finden, was Access zu bieten hat. Manches ist sicher in einem anderen Spezialtitel noch besser oder ausführlicher beschrieben. Aber Sie halten mit unserem Buch einen optimalen und überschaubaren Breitband-Mix in den Händen, der sowohl vertikal vom Einsteiger bis zum Profi als auch horizontal von den einfachen Sprachelementen bis hin zu komplexen Anwendungen jedem etwas bietet, ohne dabei den Blick auf das Wesentliche zu verlieren.

Wir hoffen, dass wir Ihnen mit diesem Buch einen nützlichen Begleiter bei der Access-Programmierung zur Seite gestellt haben, der es verdient, seinen Platz nicht im Regal, sondern griffbereit neben dem Computer einzunehmen.

Berichte

Der Bericht (*Report*) ist neben dem Formular (*Form*) die wichtigste visuelle Benutzerschnittstelle unter Access. Bei den zur Verfügung stehenden Eigenschaften, Ereignissen und Methoden gibt es viele Gemeinsamkeiten mit dem *Form*-Objekt (siehe Kapitel 4).

Bereits mit Access 2007 wurden die Berichte um eine Reihe neuer Möglichkeiten bei der Interaktion mit dem Anwender (Maus, Tastatur) erweitert. Möglich wurde dies durch zwei neue Ansichten (Berichtsansicht, Layoutansicht), deren Funktionalität in Access 2010 etwas ergänzt wurde.

Außerdem verfügt das *Report*-Objekt (im Gegensatz zum Formular) über ein recht umfangreiches Arsenal an Grafikeigenschaften und -methoden, die dem Programmierer eine Vielzahl von optischen Gestaltungsmöglichkeiten bieten.

5.1 Allgemeines

Bevor wir uns den Eigenschaften, Methoden und Ereignissen des *Report*-Objekts zuwenden, wollen wir uns zunächst allgemeineren Fragen widmen, die mit dem Öffnen eines Berichts zusammenhängen.

5.1.1 Reportansichten

Öffnen Sie einen Bericht durch Doppelklick auf seinen Namen im Navigationsbereich, so wird standardmäßig die *Berichtsansicht* angezeigt (diese Einstellung kann mittels *DefaultView*-Eigenschaft geändert werden).

- In der *Berichtsansicht* können Sie z.B. Filter anwenden, Daten kopieren oder auf Steuerelemente oder Hyperlinks klicken
- In der *Layoutansicht* können Sie das Ergebnis am Format und an weiteren Einstellungen sofort im Kontext mit den Daten betrachten, Sie können z.B. Spalten verschieben, ihre Größe verändern oder neue Felder aus der Feldliste hinzufügen
- Um einen Bericht von Grund auf neu zu erstellen verwenden Sie die *Entwurfsansicht*

- Mit Hilfe der *Seitenansicht* können Sie vor dem Drucken überprüfen, wie das Ergebnis auf dem Papier aussehen wird

5.1.2 Die OpenReport-Methode

Zum Öffnen eines Reports verwendet man die *OpenReport*-Methode des *DoCmd*-Objekts (vergleichbar mit der *OpenForm*-Methode für Formulare).

Die Syntax zeigt, dass außer dem Berichtsnamen alle weiteren Argumente optional sind:

```
DoCmd.OpenReport(ReportName, [View], [FilterName], [WhereCondition], [WindowMode], [OpenArgs])
```

- *ReportName*
Der Name des Berichts.
- *View*
Eine *AcView*-Konstante, die die Ansicht bestimmt, in welcher der Report geöffnet wird .
- *FilterName*
Namen einer gültigen Abfrage in der aktuellen Datenbank.
- *WhereCondition*
Eine gültige SQL-WHERE-Bedingung (ohne WHERE).
- *WindowMode*
Eine *AcWindowMode*-Konstante, welche den Modus angibt, in welcher das Formular geöffnet wird (*acWindowNormal*, *acHidden*, *acIcon*, *acDialog*).
- *OpenArgs*
Bestimmt die *OpenArgs*-Eigenschaft, mit welcher Parameter an den Report übergeben werden können.

Die *AcView*-Konstanten:

Konstante	Wert	Erklärung
<i>acViewNormal</i>	0	Druckansicht (Standard)
<i>acViewDesign</i>	1	Entwurfsansicht
<i>acViewPreview</i>	2	Seitenansicht (Berichtsvorschau)
<i>acViewPivotTable</i>	3	PivotTable-Ansicht
<i>acViewPivotChart</i>	4	PivotChart-Ansicht
<i>acViewReport</i>	5	Berichtsansicht
<i>acViewLayout</i>	6	Layoutansicht

BEISPIEL: Der Bericht *Mitarbeiter* wird geöffnet, alle Mitarbeiter mit dem Namen "Müller" werden angezeigt.

```
DoCmd.OpenReport "Mitarbeiter", , , "Nachname = 'Müller'"
```

5.1.3 Parameterübergabe

Das *OpenArgs*-Argument der *OpenReport*-Methode kann für die Übergabe zusätzlicher Informationen an den Bericht genutzt werden.

BEISPIEL: Zusätzlich zur WHERE-Bedingung wird ein Datum, welches in einem Textfeld steht, per *OpenArgs* an einen Bericht übergeben.

```
Dim whereCond As String
whereCond = "Anzahl > 100 AND Auslaufartikel = True"           ' WHERE-Bedingung
definieren
DoCmd.OpenReport "Bericht1", acViewPreview, , whereCond, , Text1.Value ' OpenArgs = Text1.Value
```

Anzeige des Datums im Bericht:

```
Private Sub Report_Open(Cancel As Integer)
    Bezeichnungsfld0.Caption = "Geräteliste vom " & Me.OpenArgs
    Me.FilterOn = True
End Sub
```

HINWEIS: Eine komplette Anwendung, die die verschiedenen Ansichten eines Reports demonstriert, finden Sie im Praxisbeispiel "Aufruf eines Berichts mit Datenfilter" (Seite 295).

5.2 Wichtige Berichtseigenschaften

Bei der folgenden knappen Zusammenstellung der Eigenschaften, Ereignisse und Methoden werden nur diejenigen näher erläutert, die neu gegenüber dem *Form*-Objekt (siehe Kapitel 4) sind bzw. die sich in ihrer Funktion wesentlich unterscheiden.

5.2.1 Formateigenschaften

Diese Eigenschaften bestimmen das "Outfit" eines Berichts und entsprechen größtenteils denen des *Form*-Objekts.

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
<i>AllowLayoutView</i>	Layoutansicht zulassen	True	erlaubt/verbietet Öffnen in Layoutansicht
<i>AllowReportView</i>	Berichtsansicht zulassen	True	erlaubt/verbietet Öffnen in Berichtsansicht
<i>Caption</i>	Beschriftung	Bericht	Text in Titelleiste
<i>DefaultView</i>	Standardansicht	0	0=Seitenansicht, 1=Berichtsansicht
<i>PageHeader, PageFooter</i>	Seitenkopf, Seitenfuß	0	Der Seitenkopf/-fuß wird auf allen Seiten eines Berichts gedruckt (1=außer Berichtskopf, 2=außer Berichtsfuß, außer Berichtskopf/-fuß).

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
<i>GrpKeepTogether</i>	Gruppe zusammenhalten	1	Gruppen werden pro Spalte zusammengehalten (0=pro Seite)
<i>GridX, GridY</i>	Raster X, Raster Y	10	Horizontale bzw. vertikale Einheiten des Gitternetzes (Entwurfsansicht)
<i>Height</i>	Höhe		Gesamthöhe aller Bereiche
<i>LayoutForPrint</i>	Drucklayout	<i>False</i>	Verwenden von Druckerschriftarten (<i>True</i>) oder Bildschirmschriftarten (<i>False</i>)
<i>Width</i>	Breite		Breite des Detailbereichs
<i>Picture</i>	Bild	(keines)	Pfad/Dateinamen der anzuzeigenden Bitmap
<i>PictureAlignment</i>	Bildausrichtung		Ort, wo die Hintergrundbitmap erscheinen soll (siehe <i>Form</i> -Objekt)
<i>PictureSizeMode</i>	Bildgrößenmodus		Darstellung des Hintergrundbildes (siehe <i>Form</i> -Objekt)
<i>PictureTiling</i>	Bild nebeneinander		Bilder nebeneinander anordnen
<i>PictureType</i>	Bildtyp	0	Bild wird als eingebettetes (0) oder verknüpftes (1) Objekt gespeichert
<i>PicturePages</i>	Bildseiten	0	Seite(n), auf welcher ein Bild angezeigt werden soll (alle=0, erste=1, 2=keine)

5.2.2 Dateneigenschaften

Dateneigenschaften bilden eine Untermenge der entsprechenden Eigenschaften des *Form*-Objekts.

Eigenschaft	Deutscher Bezeichner	Standard	Erläuterung
<i>RecordLocks</i>	Datensätze sperren	0	siehe <i>Form</i> -Objekt
<i>RecordSource</i>	Datenherkunft		siehe <i>Form</i> -Objekt
<i>Filter</i>	Filter		siehe <i>Form</i> -Objekt
<i>FilterOn</i>	Filter aktiv	<i>False</i>	siehe <i>Form</i> -Objekt
<i>FilterOnLoad</i>	Beim Laden filtern	<i>False</i>	siehe <i>Form</i> -Objekt
<i>OrderBy</i>	Sortiert nach		siehe <i>Form</i> -Objekt
<i>OrderByOn</i>	Sortierung aktiv	<i>True</i>	siehe <i>Form</i> -Objekt
<i>OrderOnLoad</i>	Beim Laden sortieren	<i>True</i>	siehe <i>Form</i> -Objekt

5.2.3 Grafikeigenschaften

Eigenschaften (Properties) dieser Kategorie stehen im Zusammenhang mit der Ausführung von Grafikmethoden. Für das den Grafikoperationen zugrunde liegende Koordinatensystem sind folgende Eigenschaften maßgebend:

Eigenschaft	Standardwert	Erläuterung
<i>ScaleMode</i>	1	Maßeinheit für Koordinatenangaben (Twips)
<i>ScaleWidth</i> , <i>ScaleHeight</i>		Anzahl der Einheiten für horizontale oder vertikale Innenabmessungen der Seite
<i>ScaleLeft</i> , <i>ScaleTop</i>	0, 0	Koordinaten für linke obere Ecke einer Seite
<i>CurrentX</i> , <i>CurrentY</i>	0, 0	Festlegung der aktuellen Koordinaten für Grafikmethoden

ScaleMode

Durch Ändern der *ScaleMode*-Eigenschaft können Sie sich für eine andere Maßeinheit als die standardmäßig angebotenen *Twips* entscheiden:

Wert	Bezeichnung	Erläuterung
0	User	Nutzerdefiniert, siehe <i>ScaleWidth</i> , <i>ScaleHeight</i>
1	Twips	Standardeinstellung: 1.440 Twips = 1 Inch (Zoll)
2	Point	1 Point = 0,353 mm; 72 Point = 1 Inch
3	Pixel	Entspricht Bildschirmauflösung
4	Character	1 Zeichenhöhe = 1/6 Inch; 1 Zeichenbreite = 1/12 Inch
5	Inch	1 Inch (Zoll) = 2,54 cm
6	Millimeter	1 mm = 56,7 Twips
7	Zentimeter	1 cm = 567 Twips

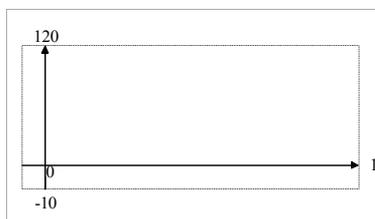
ScaleLeft und ScaleTop, ScaleWidth und ScaleHeight

Mit Hilfe dieser Eigenschaften lässt sich auf der Berichts-Oberfläche ein vollständiges Koordinatensystem mit positiven und negativen Koordinaten einrichten. Standardmäßig verläuft die y-Koordinate von oben nach unten. Durch geeignete Skalierung können Sie aber auch die gewohnte, von unten nach oben verlaufende y-Achse einstellen.

BEISPIEL: Die Anweisungen:

```
ScaleWidth = 140
ScaleHeight = 130
ScaleLeft = - 10
ScaleTop = - 120
```

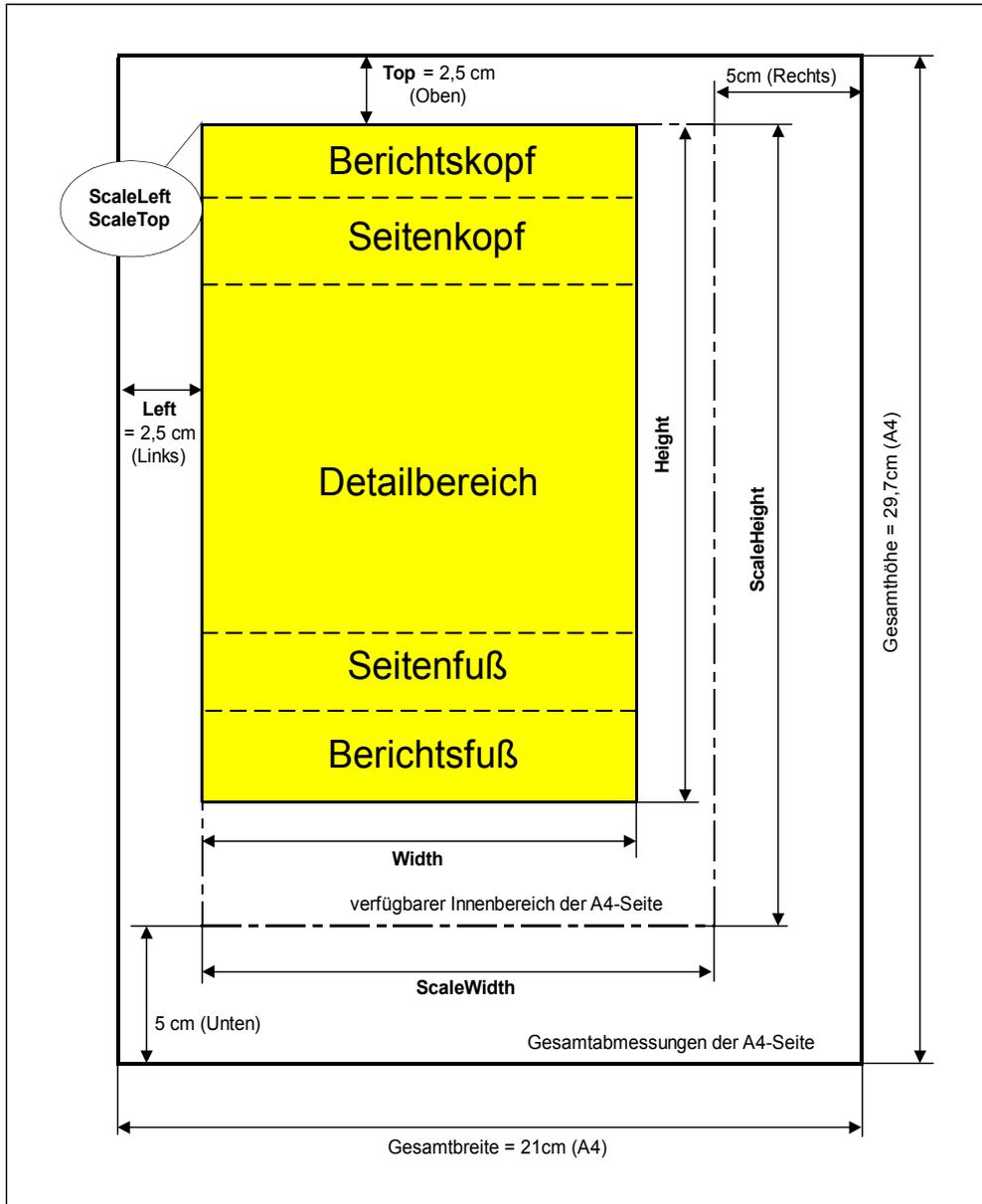
definieren das in der folgenden Abbildung gezeigte Koordinatensystem.



Left, Top, Width und Height

Die folgende Abbildung zeigt die Verhältnisse für das *Page*-Ereignis.

Hier entsprechen *Left* und *Top* dem linken und dem oberen Randabstand, wie er im Dialogfeld zum Menübefehl *Seite einrichten/Seitenlayout/Seite einrichten* festgelegt ist. *Width* und *Height* beziehen sich auf die Gesamtabmessungen des vom Berichts- und Seitenkopf, Detailbereich und Seiten- und Berichtsfuß definierten Gebietes, wie es beim Entwurf aufgezoomt werden kann.



HINWEIS: Verwechseln Sie *ScaleLeft*, *ScaleTop*, *ScaleHeight* und *ScaleWidth* nicht mit den Eigenschaften *Left*, *Top*, *Height* und *Width*!

Die *Scale*-Eigenschaften beziehen sich ausschließlich auf das innere Koordinatensystem des druckbaren Innenbereichs der Berichtsseite und auf die mit *ScaleMode* festgelegte Maßeinheit. Hingegen ist die Bedeutung von *Left*, *Top*, *Width* und *Height* davon abhängig, ob sie im *Page*-Ereignis des *Report*-Objekts oder im *Print*- bzw. *Format*-Ereignis eines seiner Bereiche (Detailbereich, Seitenkopf/-fuß, Berichtskopf/-fuß) stehen.

HINWEIS: *Left*, *Top*, *Width* und *Height* werden immer in *Twips* interpretiert, sind also unabhängig von der *ScaleMode*-Einstellung!

BEISPIEL: Der folgende Event-Handler liefert die kommentierten Ergebnisse, wenn die Seitenränder des Berichts durchgängig auf 2,5 cm eingestellt wurden (A4-Format) und Kopf-, Fuß- und Detailbereich die beim Entwurf festgelegten Abmessungen haben.

```
Private Sub Report_Page()
  ScaleMode = 7 ' Zentimeter
  Print Left / 567 ' linker Rand = 2,5 cm
  Print Top / 567 ' oberer Rand = 2,5 cm
  Print Height / 567 ' Gesamthöhe Kopf-/Fuß-/Detailbereich = 2 cm
  Print Width / 567 ' Breite Detailbereich = 10 cm
  Print ScaleLeft ' X-Verschiebung des Koordinatensystems = 0
  Print ScaleTop ' Y- " " " " = 0
  Print ScaleHeight ' Höhe des druckbaren Bereichs = 24,7 cm
  Print ScaleWidth ' Breite des druckbaren Bereichs = 16 cm
End Sub
```

Sie können die Korrektheit überprüfen, indem Sie zum Beispiel zu *ScaleWidth* die linke und die rechte Randbreite addieren. Als Ergebnis erhalten Sie im vorliegenden Fall 21 cm (DIN A4-Breite).

HINWEIS: Wenn Sie nicht innerhalb des *Report_Page*-, sondern innerhalb des *Format*- bzw. *Print*-Events zeichnen wollen, bezieht sich *Height* immer nur auf die Höhe des entsprechenden Bereichs (Detailbereich, Berichtskopf etc.), und der Koordinatenursprung (0,0) liegt in der linken oberen Ecke des Bereichs.

Bemerkungen

- Statt der Eigenschaften *ScaleHeight*, *ScaleWidth*, *ScaleLeft* und *ScaleTop* könnten Sie auch die *Scale*-Methode verwenden. Allerdings eignet sich diese nur zum Zuweisen, nicht aber zum Lesen der Abmessungen eines Koordinatensystems.
- Wenn Sie für *ScaleMode* einen Wert größer als 0 einstellen, werden die Eigenschaften *ScaleHeight* und *ScaleWidth* in die neue Maßeinheit geändert. Für *ScaleLeft* und *ScaleTop* wird automatisch der Wert 0 eingestellt. Darüber hinaus ändern sich auch die Einstellungen der Eigenschaften *CurrentX* und *CurrentY*.

CurrentX und CurrentY

Mit diesen Eigenschaften stellen Sie die aktuelle Druckposition für Grafikmethoden ein, wie es das folgende Beispiel zeigt.

BEISPIEL: Etwa in die Mitte der untersten Zeile eines A4-Blattes wird der Text "Hallo" ausgedruckt.

```
Private Sub Report_Page()
  Me.ScaleMode = 7          ' Zentimeter
  CurrentX = 8
  CurrentY = 24
  Print "Hallo"
End Sub
```

HINWEIS: Beachten Sie, dass sich die aktuelle Druckposition (*CurrentX/CurrentY*) nach Abschluss einer Grafikoperation ebenfalls verschoben hat.

5.2.4 Linien- und Stifteigenschaften

Damit legen Sie die Art der Linien fest, mit denen die Grafikmethoden *Line*, *Circle* oder *PSet* zeichnen.

Eigenschaft	Bezeichnung	Standard	Erläuterung
<i>DrawWidth</i>	Linienbreite	1	Stärke der gezogenen Linien
<i>DrawStyle</i>	Linientyp	0	Nur für <i>DrawWidth</i> = 1
<i>DrawMode</i>	Zeichnungsmodus	13	Art der Verknüpfung mit dem Hintergrund (<i>CopyPen</i>)

DrawWidth und DrawStyle

DrawWidth gibt die Breite einer Linie in Pixel an. Die Voreinstellung ist 1, was einer Breite von 1 Pixel entspricht. Ist der Wert größer als 1, erzeugen die Einstellungen 1 bis 4 für *DrawStyle* eine durchgezogene Linie. Wenn Sie für die *DrawWidth* den Wert 1 wählen, können Sie durch Zuweisen von *DrawStyle* die in der folgenden Übersicht enthaltenen Linienarten für die Methoden *Line* und *Circle* erhalten:

0	—————
1	— — — — —
2	· · · · ·
3	- - - - -
4	- · - · -
5	—————
6	—————

DrawMode

Mit dieser Eigenschaft (1...16) können Sie festlegen, ob und wie sich die für Grafikmethoden verwendete Farbe mit der Hintergrundfarbe des Berichts zur resultierenden Linienfarbe kombiniert. Standardmäßig ist der Wert 13 eingestellt (*CopyPen*), d.h., die Stiftfarbe entspricht der *ForeColor*-Property. Eine Beschreibung der übrigen Einstellungen finden Sie im Übersichtsteil dieses Kapitels (Seite 294).

5.2.5 Schrifteigenschaften

Diese Eigenschaften beziehen sich auf die *Print*-Methode.

Eigenschaft	Bezeichnung	Standardwert	Erläuterung
<i>FontBold</i>	Fettschrift	False	Nicht fett
<i>FontItalic</i>	Kursivschrift	False	Nicht kursiv
<i>FontName</i>	Schriftart		Abhängig von Systemschriftarten
<i>FontSize</i>	Schriftgröße	8	1 Punkt = 0,0353cm (1/72 Zoll)
<i>FontStrikeThru</i>	Durchgestrichen	False	Nicht durchgestrichen
<i>FontUnderline</i>	Unterstrichen	False	Nicht unterstrichen

5.2.6 Farb- und Mustereigenschaften

Für die Ausgabe von Farben und Mustern sind folgende Eigenschaften von Bedeutung:

Eigenschaft	Bezeichnung	Standard	Bemerkung
<i>ForeColor</i>	Textfarbe	Schwarz	Auch Linien- bzw. Rahmenfarbe für Rechtecke und Kreise
<i>FillColor</i>	Füllfarbe	Schwarz	Füllfarbe für Rechtecke und Kreise
<i>FillStyle</i>	Füllmuster	Transparent	Füllmuster für Rechtecke und Kreise
<i>BackColor</i>	Hintergrundfarbe	Weiß	&HFFFFFF

ForeColor und BackColor

Die Eigenschaft *ForeColor* gibt die Farbe an, welche die Methoden *Print*, *Line* und *Circle* für die Ausgabe verwenden. Wenn Sie einen Farbwert in einer Variablen speichern wollen, brauchen Sie den Datentyp *Long*.

BEISPIEL: Die folgende Sequenz druckt "Hallo" in roter Schrift:

```
Dim rot As Long
rot = vbRed
Me.ForeColor = rot
Me.Print "Hallo"
```

Die Eigenschaft *BackColor* können Sie auf bestimmte Bereiche des Berichts anwenden.

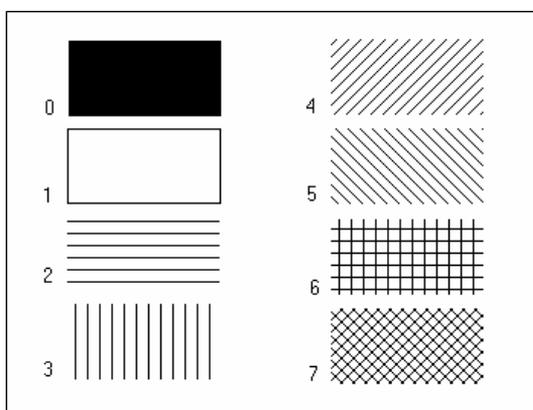
BEISPIEL: Die folgende Sequenz färbt den Detailbereich gelb.

```
Private Sub Report_Open(Cancel As Integer)
  Me.Section(0).BackColor = QBColor(14)
End Sub
```

FillColor und FillStyle

Die *FillColor*-Eigenschaft wird nur berücksichtigt, wenn die *FillStyle*-Property nicht transparent ist (also ungleich der Voreinstellung 1!). Die Farbe des Füllmusters hängt von *FillColor* ab.

Die möglichen Werte für *FillStyle* entnehmen Sie bitte der Abbildung.



QBColor- und RGB-Funktion

Insgesamt 16 Grundfarben lassen sich mit der *QBColor*-Funktion erzeugen.

Code	Farbe	Code	Farbe	Code	Farbe	Code	Farbe
0	Schwarz	4	Rot	8	Dunkelgrau	12	Hellrot
1	Blau	5	Magenta	9	Hellblau	13	Hellmagenta
2	Grün	6	Ocker	10	Hellgrün	14	Hellgelb
3	Zyan	7	Hellgrau	11	Hellzyan	15	Weiß

BEISPIEL: Die Hintergrundfarbe eines Bildfeldes erhält einen zufälligen Wert:

```
Bild1.BackColor = QBColor(Rnd * 15)
```

Wesentlich feinere Farbabstufungen lassen sich mit der *RGB*-Funktion erreichen, mit welcher Sie aus den drei Grundfarben Rot, Grün und Blau eine beliebige Farbe mischen können.

```
farbe = RGB (rot, grün, gelb)
```

Die Argumente *rot*, *grün*, *blau* können Integer-Werte zwischen 0 und 255 annehmen.

Die folgende Tabelle enthält einige Standardfarben mit den zugehörigen Rot-, Grün- und Blau-Anteilen:

Farbe	Rot	Grün	Blau
Schwarz	0	0	0
Blau	0	0	255
Grün	0	255	0
Zyan	0	255	255
Rot	255	0	0
Magenta	255	0	255
Gelb	55	255	0
Weiß	255	255	255

BEISPIEL: Verwendung von *RGB*:

```
rot = RGB(255,0,0)
schwarz = RGB(0,0,0)
```

Farbkonstanten

Für eine Vielzahl von Farben können auch die in VBA integrierten Farbkonstanten wie *vbRed*, *vbWhite* etc. Verwendung finden.

BEISPIEL: Eine rote Stiftfarbe wird eingestellt:

```
Me.ForeColor = vbRed
```

HINWEIS: Eine Zusammenstellung wichtiger Farbkonstanten finden Sie im Übersichtsteil des Kapitels (Seite 294).

5.2.7 Sonstige Eigenschaften

Im folgenden Sammelsurium treffen Sie hauptsächlich solche Eigenschaften an, die es in äquivalenter Bedeutung bereits beim Formular gibt (siehe Kapitel 4).

Eigenschaft	Bezeichnung	Erläuterung
<i>DateGrouping</i>	Datumsgruppierung	Datumsangaben (0 = amerikanisch, 1 = deutsch)
<i>FastLaserPrinting</i>	Schneller Laserdruck	Siehe <i>Form</i> -Objekt
<i>HelpFile</i>	Hilfedatei	Siehe <i>Form</i> -Objekt
<i>HelpContextID</i>	Hilfekontext	Siehe <i>Form</i> -Objekt
<i>Section(n)</i>	Bereich	Siehe Erläuterung unten
<i>Tag</i>	Marke	Siehe <i>Form</i> -Objekt

Section

In Analogie zum *Form*-Objekt werden die einzelnen Bereiche eines Berichts durch die *Section*-Eigenschaft (schreibgeschützt) des *Report*-Objekts dargestellt.

Section kann auch als Objekt (quasi als Control) betrachtet werden. Dies ist kein Widerspruch, denn Eigenschaften können bekanntlich wiederum Objekte sein.

Der Index der *Section*-Eigenschaft ist ein *Integer*-Wert, der einen bestimmten Bereich bezeichnet:

Index	Konstante	Beschreibung
0	<i>acDetail</i>	Detailbereich
1	<i>acHeader</i>	Berichtskopfbereich
2	<i>acFooter</i>	Berichtsfußbereich
3	<i>acPageHeader</i>	Seitenkopfbereich
4	<i>acPageFooter</i>	Seitenfußbereich
5	<i>acGroupLevel1Header</i>	Gruppenebene 1 Kopfbereich
6	<i>acGroupLevel1Footer</i>	Gruppenebene 1 Fußbereich
7	<i>acGroupLevel2Header</i>	Gruppenebene 2 Kopfbereich
8	<i>acGroupLevel2Footer</i>	Gruppenebene 2 Fußbereich

Section hat wiederum zahlreiche Eigenschaften (*BackColor*, *AlternateBackColor*, *Visible*, *DisplayWhen* etc.), die man sich am besten anhand von Beispielen verdeutlicht:

BEISPIEL: Der Fußbereich eines Berichts (oder Formulars) wird ausgeblendet:

```
Me.Section(acFooter).Visible = False
```

BEISPIEL: Die Anweisung:

```
Me.Section(acDetail).AlternateBackColor = vbYellow
```

stellt alternierende Zeilenfarbe für den Detailbereich des Berichts (oder Formulars) ein.

HINWEIS: Auch für Steuerelemente können Sie die *Section*-Eigenschaft verwenden, um zu ermitteln, in welchem Bereich eines Formulars oder Berichts sich das Steuerelement befindet.

BEISPIEL: Die Variable *i* wird auf den Wert 2 gesetzt, falls sich das Textfeld *Text30* im Berichtsfußbereich (oder Formularfußbereich) befindet.

```
Dim i As Integer
i = Me!Text30.Section
```

5.3 Berichtereignisse

Obwohl das eigentliche Medium eines Berichts das Papier und nicht der Bildschirm ist, gibt es bezüglich der Anzahl der Ereignisse (Events) kaum noch Unterschiede zwischen einem Report und einem Formular¹.

5.3.1 Allgemeine Ereignisse

Die folgende Tabelle zeigt wichtige *Report*-Ereignisse:

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
<i>Open</i> (Cancel As Integer)	Beim Öffnen
<i>Close</i> ()	Beim Schließen
<i>Activate</i> ()	Bei Aktivierung
<i>Deactivate</i> ()	Bei Deaktivierung
<i>NoData</i> (Cancel As Integer)	Bei Ohne Daten
<i>Page</i> ()	Bei Seite
<i>Error</i> (DataErr As Integer, Response As Integer)	Bei Fehler
<i>ApplyFilter</i>	Bei angewendetem Filter
<i>Filter</i>	Bei Filter
<i>Load</i>	Bei Laden
<i>Unload</i>	Bei Entladen
<i>GotFocus</i>	Bei Fokuserhalt
<i>LostFocus</i>	Bei Fokusverlust
<i>Resize</i>	Bei Größenänderung
<i>Timer</i>	Bei Zeitgeber

Detailbereich, Kopf- und Fußzeile für die Seite bzw. für einen Bericht haben andere Ereignisse:

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
<i>Retreat</i> ()	Bei Rücknahme (nur Detailbereich und Berichtskopf/-fuß)
<i>Print</i> (Cancel As Integer, PrintCount As Integer)	Beim Drucken
<i>Format</i> (Cancel As Integer, FormatCount As _ Integer)	Beim Formatieren
<i>Paint</i>	Beim Anzeigen

¹ Das betrifft vor allem die Preview- und die Layout-Ansicht des Reports.

Open, Close, Activate, Deactivate, Error

Diese Ereignisse sind äquivalent zu den gleichnamigen Formular-Ereignissen. Dies betrifft auch die Reihenfolge, in der sie auftreten (siehe Kapitel 4).

Page, NoData, Print, Format

Das *Page*-Ereignis wird ausgelöst, nachdem Access die Seite eines Berichts zum Drucken formatiert hat, jedoch bevor die Seite gedruckt wird. Dieses Ereignis wird verwendet, um einen Rahmen für die Seite zu erstellen oder um der Seite andere grafische Elemente hinzuzufügen.

Das *NoData*-Ereignis wird vor dem erstmaligen Auftreten des *Page*-Events ausgelöst. Es tritt ein, nachdem Microsoft Access einen Bericht zum Drucken formatiert hat, der keine Daten enthält (leere Datensatzgruppe), aber bevor der Bericht gedruckt wird. Man verwendet dieses Ereignis um zu verhindern, dass ein leerer Bericht gedruckt wird.

Das *Print*-Ereignis tritt auf, wenn Daten in einem Berichtsbereich zum Drucken formatiert werden, jedoch bevor der Bereich gedruckt wird. Bei *Detailbereichen* wird *Print* für jeden Datensatz im Bereich ausgelöst, unmittelbar bevor die Daten gedruckt werden. In Gruppenköpfen/-füßen tritt *Print* bei jeder neuen Gruppe auf. Hier könnten Sie z.B. Seitensummen berechnen, die in die Kopf- oder Fußzeile gedruckt werden.

Das *Format*-Ereignis wird ausgelöst, wenn Access die Daten ermittelt, die in einen Berichtsbereich gehören, jedoch bevor der Bereich für die Vorschau oder für das Drucken formatiert wird. Das *Format*-Ereignis verwenden Sie auch bei Änderungen, die das Seitenlayout betreffen, wie z.B. das Ein-/Ausblenden von Steuerelementen oder wenn Sie auf Daten aus Bereichen zugreifen müssen, die nicht gedruckt werden (z.B. wenn Sie eine fortlaufende Summe berechnen, aber nur einzelne Seiten des Berichts drucken wollen).

BEISPIEL: Das aktuelle Datum wird fett formatiert an den durch *CurrentX* und *CurrentY* festgelegten Koordinaten ausgegeben:

```
Sub ReportHeader0_Print (Cancel As Integer, PrintCount As Integer)
    Me.FontBold = True
    Me.Print Date()
End Sub
```

BEISPIEL: In der Ereignisprozedur *Detailbereich_Print* wird die Farbe des Rechteck-Hintergrundes nach jeder ausgedruckten Zeile geändert:

```
Dim frb As Boolean

Private Sub Detailbereich_Print(Cancel As Integer, PrintCount As Integer)
    If frb Then
        Rechteck.BackColor = vbYellow
    Else
        Rechteck.BackColor = vbWhite
    End If
    frb = Not frb
End Sub
```

5.3.2 Tastatur- und Mausereignisse

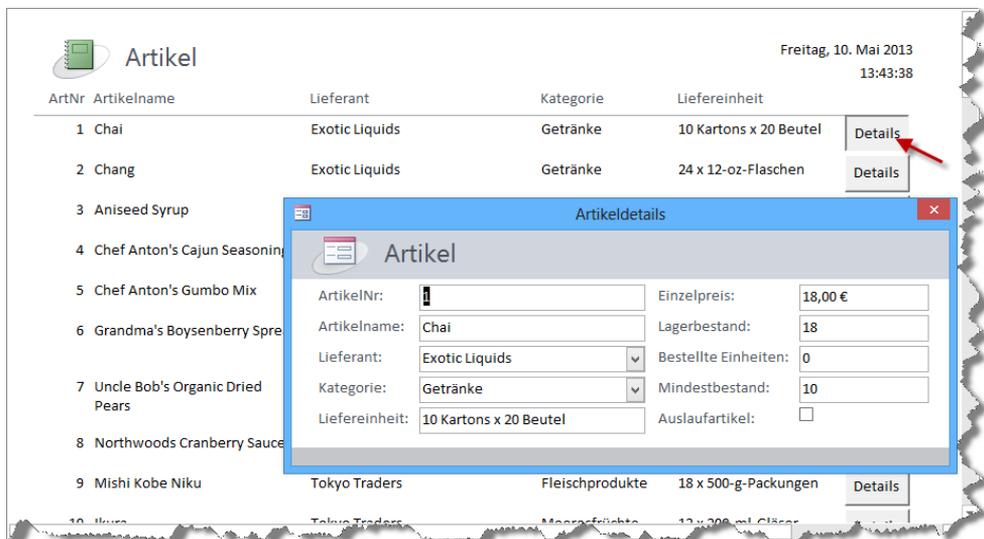
Die meisten der im Folgenden aufgelisteten Ereignisse des *Report*-Objekts treffen nur für die Berichtsansicht (manchmal auch Layoutansicht) zu.

Ereignisroutine (Event-Handler)	Ereigniseigenschaft (deutsch)
KeyDown (KeyCode As Integer, Shift As Integer)	Bei Taste Ab
KeyPress (KeyAscii As Integer)	Bei Taste
KeyUp (KeyCode As Integer, Shift As Integer)	Bei Taste Auf
<i>MouseDown</i>	Bei Maustaste Ab
<i>Click</i>	Beim Klicken
<i>DbClick</i>	Beim Doppelklicken
<i>MouseMove</i>	Bei Mausbewegung
<i>MouseUp</i>	Bei Maustaste Auf
<i>MouseWheel</i>	Bei Mausrad

BEISPIEL: Um das *Click*-Ereignis zu demonstrieren, lassen wir uns mit Assistentenhilfe einen Bericht für die *Artikel*-Tabelle der *Nordwind*-Datenbank generieren und fügen in den Detailbereich ganz rechts eine Befehlsschaltfläche *Befehl1* ein (damit diese Schaltfläche nicht mit ausgedruckt wird, setzen wir deren *Anzeigen*-Eigenschaft auf *Nur am Bildschirm*). Der *Click*-Eventhandler unterscheidet sich nicht von dem eines Formulars:

```
Private Sub Befehl1_Click()
    DoCmd.OpenForm "Artikeldetails", acNormal, , "ArtikelNr=" & [ArtikelNr], , acDialog
End Sub
```

Nach Öffnen des Berichts genügt ein Klick auf eine bestimmte Schaltfläche, um das Dialogfeld *Artikeldetails* zu öffnen.



5.4 Berichtsmethoden

Ein wesentlicher Unterschied zwischen Report und Formular ist, dass man auf die Oberfläche eines Reports auch zeichnen kann. Der überwiegende Teil der Methoden eines Reports sind deshalb Grafikmethoden.

5.4.1 Grafikmethoden (Übersicht)

Die im Folgenden erörterten Grafikmethoden werden grundsätzlich nur im *Page*-Ereignis des Reports bzw. im *Print*- bzw. *Format*-Ereignis eines seiner Bereiche ausgeführt.

Methodenname	Erklärung
Print <i>Ausgabeliste</i>	Druckt Text
Line <i>[[Step](x1, y1)] – [Step](x2, y2)[, [Farbe][, B[F]]]</i>	Zeichnet Linie oder Rechteck
PSet <i>[Step](x, y)[, Farbe]</i>	Setzt einen Punkt
Circle <i>[Step](x, y, Radius)[, [Farbe][, [Startwert] [, [Endwert][, Seitenverhältnis]]]</i>	Zeichnet Kreis oder Kreissegment
TextHeight (<i>Zeichenfolge</i>)/ TextWidth (<i>Zeichenfolge</i>)	Gibt Höhe/Breite einer Textzeichenfolge zurück
Scale <i>[(x1, y1) – (x2, y2)]</i>	Definiert Koordinatensystem

5.4.2 Scale

Mit dieser Methode können Sie die umständliche Definition eines Koordinatensystems mit den Eigenschaften *ScaleLeft*, *ScaleTop*, *ScaleWidth* und *ScaleHeight* vermeiden.

Syntax:

```
Scale [(x1, y1) - (x2, y2)]
```

Argumente im Einzelnen:

x1, y1 linke obere Ecke des Koordinatensystems

x2, y2 rechte untere Ecke des Koordinatensystems

BEISPIEL: Die Anweisung:

```
Scale (-10, 120)-(130, -10)
```

führt zum gleichen Koordinatensystem:

```
ScaleWidth = 140
ScaleHeight = 130
ScaleLeft = - 10
ScaleTop = 120
```

HINWEIS: Die *Scale*-Methode ohne Argumente setzt das Koordinatensystem auf die Maßeinheit *Twips* (Standard)!

5.4.3 Line

Diese Methode zieht eine gerade Linie.

Syntax:

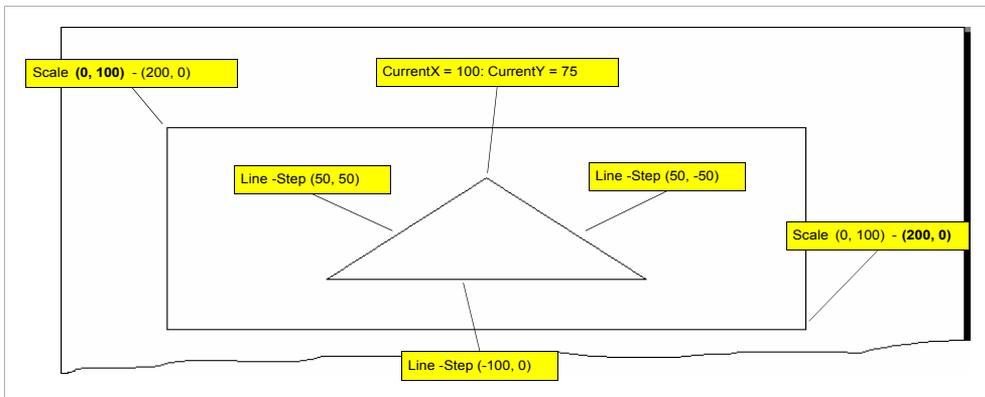
```
Line [[Step1](x1, y1)] - [Step2](x2, y2)[, [Farbe][, B[F]]]
```

Die Argumente in der umfangreichen Parameterliste haben folgende Bedeutung:

- Step1* Der Parameter gibt an, dass die Koordinaten des Anfangspunkts relativ zur aktuellen Grafikkposition liegen, die durch *CurrentX* und *CurrentY* festgelegt ist.
- x1, y1* Dies sind *Single*-Werte, welche die Koordinaten des Anfangspunktes der Linie oder der linken oberen Ecke des Rechtecks bezeichnen. Die Eigenschaften *ScaleMode*, *ScaleLeft*, *ScaleTop*, *ScaleHeight* und *ScaleWidth* legen die verwendete Maßeinheit fest. Werden *x1,y1* nicht angegeben, so beginnt die Linie an der von *CurrentX* *CurrentY* definierten Position.
- Step2* Gibt an, dass die Endpunktkoordinaten relativ zu *x1,y1* liegen.
- x2, y2* Dies sind *Single*-Werte für die Endpunktkoordinaten der zu zeichnenden Linie.
- farbe* Legt die Farbe für die zu zeichnende Linie fest (*Long*-Wert). Wenn dieses Argument fehlt, wird die *ForeColor*-Eigenschaft verwendet. Sie können für dieses Argument auch direkt die Funktionen *RGB* oder *QBColor* einsetzen.
- B* Bewirkt, dass ein Rechteck gezeichnet wird. Die angegebenen Koordinaten bestimmen dabei die linke obere und die rechte untere Ecke des Rechtecks.
- F* Kann nur zusammen mit der Option *B* verwendet werden und gibt an, dass das Rechteck mit der Zeichenfarbe ausgefüllt werden soll. Bei *B* ohne *F* wird das Rechteck mit der Farbe und dem Muster ausgefüllt, die durch die Eigenschaften *FillColor* und *FillStyle* festgelegt sind.

BEISPIEL: Der folgende Code im *Print*-Ereignis des Seitenkopfes erzeugt dort ein neues Koordinatensystem (200 breit, 100 hoch, positiv verlaufende x- und y-Achse). Der Seitenkopf wird umrahmt und ein Dreieck in die Mitte gezeichnet.

```
Private Sub Seitenkopf_Print(Cancel As Integer, PrintCount As Integer)
    Scale (0, 100)-(200, 0)           ' neues Koordinatensystem
    Line (0, 100)-(200, 0), , B      ' umrahmendes Rechteck
    CurrentX = 100: CurrentY = 75    ' obere Dreieckspitze
    Line -Step(50, -50)              ' rechte Dreieckkante
    Line -Step(-100, 0)              ' untere Dreieckkante
    Line -Step(50, 50)               ' linke Dreieckkante
End Sub
```



BEISPIEL: Die Anweisung:

```
Line (0, 100)-(200, 0), QBColor(12), BF
```

würde ein rot ausgefülltes umrahmendes Rechteck erzeugen.

5.4.4 PSet

Damit setzen Sie einen einzelnen Punkt.

Syntax:

```
PSet [Step](x, y)[, Farbe]
```

Die Argumentliste entspricht im Prinzip der (verkürzten) *Line*-Methode.

BEISPIEL: Umständlich, aber lehrreich: Längs der horizontalen Achse eines Berichts wird mit *PSet* eine rote Linie gezeichnet.

```
Dim mitte As Single, i As Integer
Me.ScaleMode = 3           ' Maßeinheit: Pixel
mitte = Me.ScaleHeight / 2 ' Mittelachse
For = 1 To Me.ScaleWidth
  Me.PSet(i, mitte), QBColor(12)
Next i
```

Bemerkungen

- Die Größe eines Punktes hängt von der Eigenschaft *DrawWidth* ab. Die Art, wie er gezeichnet wird, ist abhängig von den Eigenschaften *DrawMode* und *DrawStyle*.
- Für das Löschen eines Pixels verwenden Sie die Farbe Weiß (&HFFFFFF).

5.4.5 Circle

Damit zeichnen Sie einen Kreis, eine Ellipse oder einen Bogen auf einem *Report*-Objekt.

Syntax:

```
Circle [Step](x, y), Radius[, [Farbe][, [Startwert][, [Endwert][, Seitenverhältnis]]]
```

Die Bedeutung der Argumente im Einzelnen:

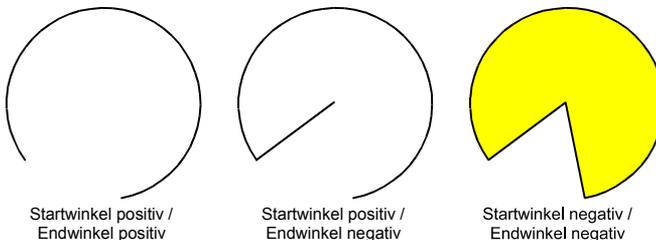
<i>Step</i>	Gibt an, dass der Mittelpunkt relativ zu den aktuellen Koordinaten liegt
<i>x, y</i>	Definiert die Koordinaten des Mittelpunktes (<i>Single</i> -Werte)
<i>radius</i>	Radius (<i>Single</i> -Wert)
<i>farbe</i>	Parameter, der die Farbe der Kreislinie angibt (<i>Long</i> -Wert). Fehlt dieses Argument, so wird <i>ForeColor</i> genommen.
<i>startwert,</i> <i>endwert</i>	Für einen Kreis- oder Ellipsenbogen geben Start- und Endwert die Anfangs- und die Endposition des Bogens an (in Radiant). Standardmäßiger Startwert ist 0 Radiant, standardmäßiger Endwert ist 2 Pi Radiant (Vollkreis).
<i>seitenverhältnis</i>	Das Seitenverhältnis der Ellipse, Standardwert ist 1.0 (Kreis).

Bemerkungen

- Um den Start- bzw. Endwert in Grad angeben zu können, muss jeweils mit $Pi/180$ multipliziert werden.

```
Const Pi = 3.1416
Dim rad As Single, grad As Single
...
rad = grad * Pi / 180
```

- Die Winkelangaben zählen immer entgegen der Uhrzeigerichtung.



- Sind Start- bzw. Endwert negativ, so wird der jeweilige Radius mit eingezeichnet (Tortenstück).
- Nach Beendigung der *Circle*-Methode sind *CurrentX* und *CurrentY* auf die Mittelpunktskoordinaten eingestellt.
- Linienart und -breite, Füllfarbe und Füllmuster sind abhängig von den Eigenschaften *Draw-Mode*, *DrawStyle*, *DrawWidth*, *FillColor* und *FillStyle*.

BEISPIEL: Ein Tortendiagramm soll genau in der Mitte einer Berichtsseite erzeugt werden.

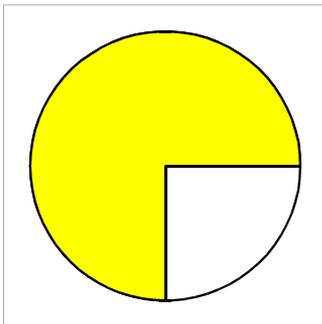
Im ersten Schritt zeichnen wir einen (leeren) Vollkreis:

```
Const Pi = 3.14159
Dim mitteX As Single, mitteY As Single, radius As Single
mitteX = Me.ScaleWidth / 2
mitteY = Me.ScaleHeight / 2
radius = Me.ScaleHeight / 4
Me.Circle (mitteX, mitteY), radius
```

Im nächsten Schritt wird die gelb gefüllte "Dreivierteltorte" erzeugt. Dazu müssen Start- und Endwinkel negative Vorzeichen haben.

```
Dim start As Single, ende As Single
start = -0.00000001
ende = -3 * Pi / 2 ' 270 Grad
Me.FillColor = QBColor(14) ' gelb
Me.FillStyle = 0 ' solide Füllung
Me.Circle (mitteX, mitteY), radius, , start, ende
```

Wenn wir diesen Code in das *Page*-Event des Reports einbauen, erscheint das Kreisdiagramm sehr groß (Durchmesser entspricht der Hälfte der Seitenhöhe, ohne Ränder). Wir können den gleichen Code aber auch in das *Print*-Event des Detailbereichs einsetzen. Dann beziehen sich die *Scale*-Eigenschaften auf die von uns eingestellte Höhe und Breite des Detailbereichs, das Kreisdiagramm wird erheblich kleiner ausfallen.



5.4.6 Print

Auch die Textausgabe zählt unter Windows mit zu den Grafikmethoden! Die *Print*-Methode gibt eine Zeichenkette (String) oder auch nur den Wert von Variablen bzw. Funktionen auf einem *Report*-Objekt (oder im *Direkt*-Fenster) aus. Dabei werden die aktuelle Farbe (*ForeColor*) und Schriftart (*Font*-Eigenschaften) verwendet.

Syntax:

```
Print {Spc(n) | Tab(n)} ausdruck zeichPos
```

Die Argumente:

<i>Spc(n)</i>	Wird optional verwendet, um n Leerzeichen auszugeben.
<i>Tab(n)</i>	Wird optional verwendet, um die Druckposition zur n-ten Spaltennummer zu bewegen. Tab ohne Argumente verschiebt die Druckposition zum Anfang des nächsten Ausgabebereichs.
<i>ausdruck</i>	Wird optional ausgedruckt (numerischer Ausdruck oder String)
<i>zeichPos</i>	Legt die Druckposition für das nächste Zeichen fest. Mehrere Ausdrücke können mit einem Semikolon bzw. einem Leerzeichen getrennt werden.

BEISPIEL: Die folgenden zwei Programmzeilen liefern ähnliche (aber nicht identische) Ergebnisse.

```
Print "Hallo"; Tab(50); "Hallo"
```

```
Print "Hallo"; Space(50); "Hallo"
```

5.4.7 TextWidth und TextHeight

Beide Methoden geben die Breite bzw. Höhe einer Textzeichenfolge zurück.

Syntax:

```
TextHeight(Zeichenfolge)
```

bzw.

```
TextWidth(Zeichenfolge)
```

Mit *TextWidth/TextHeight* werden die horizontale bzw. vertikale Ausdehnung einer Textzeichenfolge in der mit *Scale* festgelegten Maßeinteilung ermittelt, wenn der Bericht formatiert und gedruckt wird. Das Ergebnis ist abhängig von den Schriftarteigenschaften (*FontName, FontSize* etc.). Sind Wagenrücklaufzeichen enthalten, so ermittelt *TextWidth* die Breite der längsten Zeile (vom Zeilenanfang bis zum Wagenrücklaufzeichen). *TextHeight* gibt in diesem Fall die kumulierte Höhe der Zeilen zurück, inklusive des normalen Abstandes über und unter jeder Zeile.

Sie können die von den Methoden *TextWidth* und *TextHeight* zurückgegebenen Werte verwenden, um den erforderlichen Platz für mehrere Textzeilen innerhalb eines Berichts zu berechnen.

BEISPIEL: Der Event-Handler druckt die Überschrift eines Berichtes 20 Pixel unterhalb des oberen Randes zentriert aus:

```
Private Sub Report_Page()  
    Dim schrift As String  
    schrift = "Produktbericht"  
    Me.ScaleMode = 3 ' Pixel
```

```

Me.FontSize = 24
CurrentY = 20
CurrentX = (Me.ScaleWidth - Me.TextWidth(schrift)) / 2
Me.Print schrift
End Sub

```

5.4.8 Sonstige Methoden

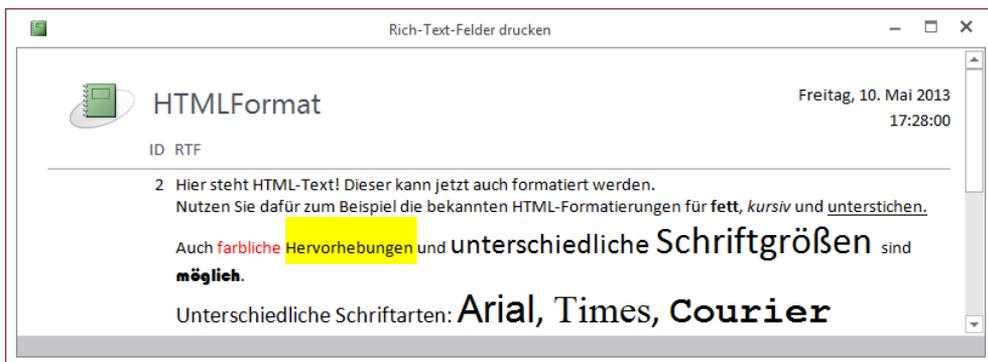
Andere Report-Methoden als die für Grafikausgaben sind sehr spärlich, erwähnenswert sind eigentlich nur *Requery* (Datenaktualisierung in einem Unterbericht) und *SetTabOrder* (Tabulatorenreihenfolge im Detailbereich).

5.5 Weitere Features des Report-Objekts

Im Folgenden wollen wir Ihnen einige speziellere Features vorstellen, wie sie im Wesentlichen erst ab Access 2007 eingeführt wurden.

5.5.1 Rich-Text-Felder drucken

Bekanntlich kann man jetzt auch HTML-Steuerzeichen für die Formatierung von Text- bzw. Memofeldern verwenden. Setzen Sie dazu die *TextFormat*-Eigenschaft auf *Rich-Text*. Zusätzlich kann man die *Vergrößerbar*-Eigenschaft (*CanGrow*) auf *True* setzen.



Sollten Sie die *TextFormat*-Eigenschaft auf *Nur-Text* gestellt haben, wird Ihnen der entsprechende HTML-Code des Datenfeldes angezeigt:

```

<div>Hier steht HTML-Text! Dieser kann jetzt auch formatiert werden. </div>

<div>Nutzen Sie dafür zum Beispiel die bekannten HTML-Formatierungen für <strong>fett</strong>,
<em>kursiv </em>und <u>unterstichen. </u></div>

<div>Auch <font color=red>farbliche </font><font
style="BACKGROUND-COLOR:#FFFF00">Hervorhebungen </font>und <font size=5>unterschiedliche
</font><font
size=6>Schriftgrößen </font>sind <font face="Bauhaus 93">möglich</font>.</div>

```

5.5.2 Verlauf eines Memofeldes drucken

Haben Sie die "Nur anfügen"-Eigenschaft eines Memofeldes aktiviert, werden alle Änderungen an diesem Feld mit Datumstempel dokumentiert. Da dies kein reiner Selbstzweck ist, wollen Sie den Verlauf (bzw. die Historie) zusammen mit dem aktuellen Inhalt ausdrucken.

Hier hilft Ihnen die *Application.ColumnHistory*-Methode weiter:

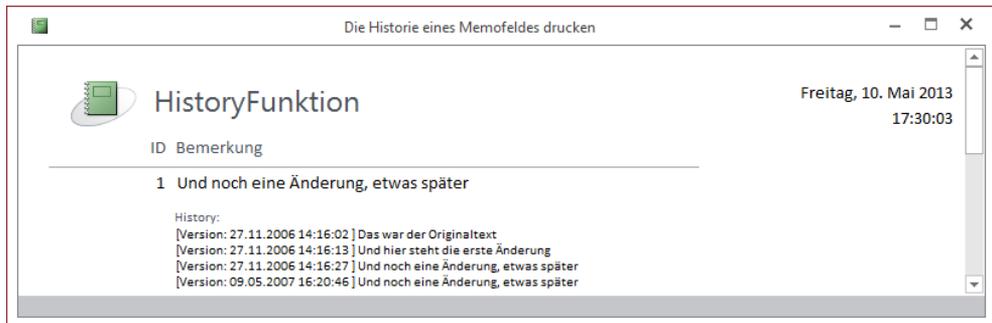
```
ColumnHistory(Tabellenname, SpaltenName, Suchstring)
```

Tabellenname und *Spaltenname* dürften selbsterklärend sein, der *Suchstring* entspricht einer SQL WHERE-Klausel (ohne WHERE) bzw. dem Suchstring bei einem *Find*. Hier können Sie also den Primärschlüssel zur Suche einsetzen.

Entwerfen Sie wie gewohnt den Bericht, fügen Sie ein zusätzliches *Textfeld* ein und legen Sie dessen *Steuerelementinhalt* wie folgt fest:

```
=ColumnHistory([Datensatzquelle];"Bemerkung";"[ID]=" & Nz([ID];0))
```

Starten Sie den Bericht in der Druckvorschau und schnüffeln Sie in der "Vergangenheit" herum:



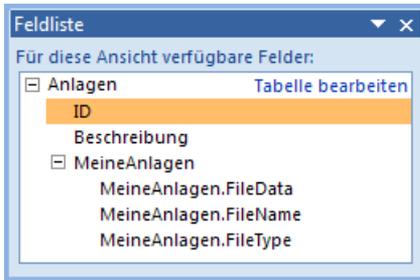
5.5.3 Eine Liste der Anlagen drucken

Seit Access 2007 besteht die Möglichkeit, Bilder, Dokumente etc. als Anlage in einer Tabelle zu speichern. Da Sie auch mehrere Anlagen pro Feld verwalten können, ist es sicher interessant, die Liste der enthaltenen Anlagen zu ermitteln.

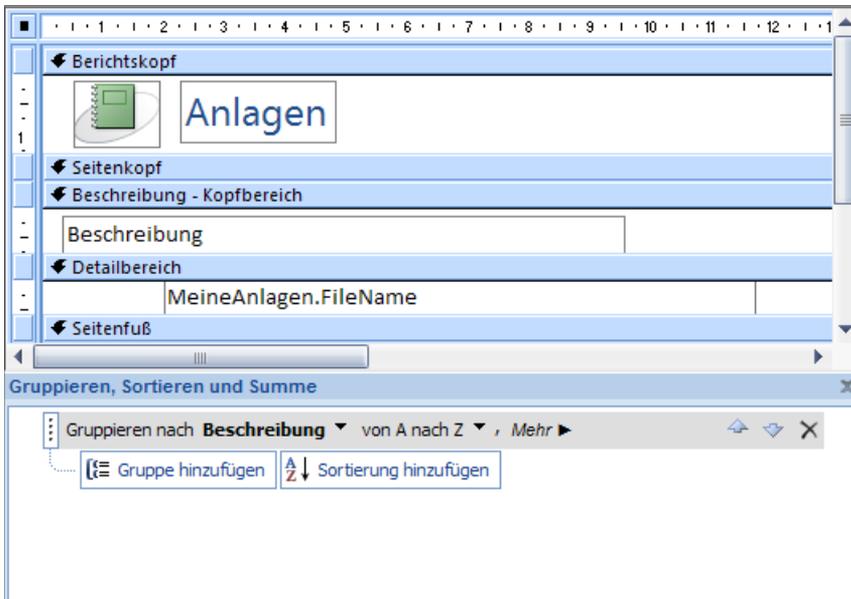
Variante 1 (Gruppierung)

Den Zugriff auf den Inhalt des Anlagefeldes erhalten Sie über ein geschachteltes Recordset (quasi die Detaildatensätze). Dieses wird auch in der Liste der verfügbaren Datenfelder angezeigt und kann für die Gestaltung des Reports genutzt werden (siehe folgende Abbildung).

Ziehen Sie einfach das Feld *FileName* in den Detailbereich des Berichts und Sie erhalten eine Liste aller Anlagen. Allerdings wird in diesem Fall der übergeordnete Datensatz mehrfach aufgelistet.



Um dies zu verhindern, gruppieren Sie die Daten einfach nach der ID des übergeordneten Datensatzes bzw. eines anderen Feldes.



Der resultierende Bericht:



Variante 2 (VBA)

Wollen oder können Sie die Daten nicht gruppieren, bleibt Ihnen noch der Weg über ein zusätzliches *Recordset*. Mit diesem können Sie direkt auf die einzelnen Einträge des Anlage-Feldes zugreifen, wie es das folgende kleine Beispiel beschreibt.

BEISPIEL: Auslesen der Anlage-Feld-Details

Option Explicit

Ein extra Recordset definieren:

```
Private rst As Recordset
```

Das Recordset wird mit dem Öffnen des Reports initialisiert (gleiche Daten wie der Report):

```
Private Sub Report_Open(Cancel As Integer)
    Set rst = CurrentDb().OpenRecordset("Anlagen", dbOpenTable)
    rst.Index = "PrimaryKey"
End Sub
```

Mit der folgenden Funktion wird zunächst der passende Datensatz gesucht und anschließend ein Recordset für die Detaildaten (Anlagefeld) erzeugt.

```
Private Function GetAnlagen() As String
    Dim rsta As Recordset
    Dim s As String
    Dim i As Integer

    s = ""
    rst.Seek "=", Id.value
    Set rsta = rst.Fields("MeineAnlagen").value
```

Mit diesem Recordset können Sie die einzelnen Anlagen abarbeiten (wir erzeugen eine Textliste):

```
Do While Not rsta.EOF
    s = s & Format(i, "00 : ") & rsta!filename & vbCrLf
    rsta.MoveNext
    i = i + 1
Loop
rsta.Close
Set rsta = Nothing
GetAnlagen = s
End Function
```

Alles was nun noch bleibt, ist das Zuweisen der obigen Funktion als *Steuerelementinhalt* eines Textfeldes:

```
=GetAnlagen()
```

Die folgende Abbildung zeigt das Endergebnis:

Anlagen	
ID	Beschreibung:
1	Hier ist der erste Datensatz
2	Hier ist der zweite Datensatz

Anlagen:	
00	: Autumn Leaves.jpg
01	: Creek.jpg
02	: Humpback Whale.jpg
00	: Green Sea Turtle.jpg
01	: Waterfall.jpg

5.5.4 Berichte nachträglich filtern

Nachdem Sie einen Bericht entworfen haben, können Sie die getroffene Auswahl der Datensätze ändern bzw. weiter einschränken.

Variante 1 (IDE)

Mit der Version 2007 wurden auch erweiterte Filteroptionen in der Layout- bzw. der Berichtsansicht implementiert, die Sie unter anderem über die Kontextmenüs der einzelnen Datenfelder erreichen.

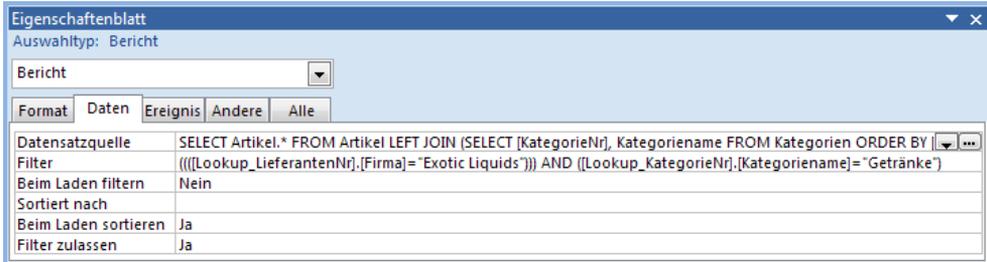
Wählen Sie beispielsweise den Filter "Ist gleich Getränke" wird dieser sofort auf die aktuelle Ansicht angewendet. Auf diese Weise können Sie einzelnen Spalten neue Filter hinzufügen bzw. diese über den entsprechenden Menüpunkt "Filter löschen aus [Spaltenname]" auch wieder deaktivieren.

Kategorie	Liefereinheit
Getränke	10 Kartons x 20 Beutel
Geträn	en
Gewür	nen
Gewür	
Gewür	
Gewür	
Naturp	en
Gewür	
Fleisch	ngen
Meere	er

Ausschneiden
Kopieren
Einfügen
Von A bis Z sortieren
Von Z bis A sortieren
Filter löschen aus KategorieNr
Textfilter
Ist gleich "Getränke"
Ist ungleich "Getränke"
Enthält "Getränke"
Enthält nicht "Getränke"

Variante 2 (VBA)

Dass derartige Filter auch aus VBA-Sicht problemlos realisierbar sind, dürfte nach einem Blick in die Eigenschaften des Berichts schnell klar werden:



Es wird lediglich der *Filter*-Eigenschaft ein entsprechender Ausdruck zugewiesen, was wir auch aus dem Programm heraus realisieren können.

BEISPIEL: Filtern von Daten über eine Schaltfläche im Berichtskopf

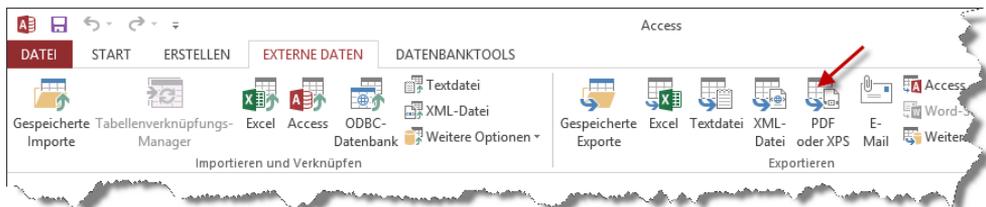
```
Private Sub Befeh124_Click()
    Me.Filter = "([Lookup_LieferantenNr].[Firma]="Exotic Liquids")"
    Me.FilterOn = True
End Sub
```

Bemerkungen

- Möchten Sie dieses Feature den Nutzern Ihrer Berichte vorenthalten, was bei vordefinierten Reports durchaus sinnvoll ist, können Sie die Berichtseigenschaft "Filter zulassen" auf *False* setzen.
- Setzen Sie die Eigenschaft "Beim Laden filtern" auf *False*, andernfalls kann es schnell passieren, dass ein bereits vergessener Filter noch aktiv ist.

5.5.5 Berichte als PDF-Datei exportieren

Access bietet auch eine integrierte Funktionalität für den PDF-Export. Über das Menüband (*Externe Daten/Exportieren/PDF oder XPS*) steht Ihnen die gewünschte Funktion zur Verfügung.

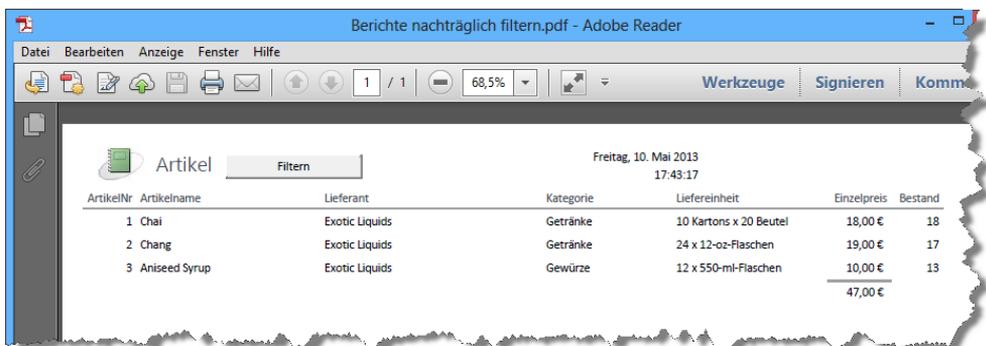


Alternativ können Sie den Export jedoch auch per Visual Basic-Code auslösen:

BEISPIEL: Export im PDF-Format

```
Sub ExportAlsPDF()
    DoCmd.OutputTo acOutputReport, "Berichte nachträglich filtern", acFormatPDF, "c:\Test.pdf", _
        True
End Sub
```

Nach dem Export wird, wenn vorhanden, der Acrobat-Reader aufgerufen:



5.5.6 Berichte als RTF-Datei exportieren

Gleich einen Dämpfer vorweg: Dieses Feature funktioniert nur rudimentär, viele Formatierungen werden in der exportierten RTF-Datei nicht oder nur unvollständig wiedergegeben.

BEISPIEL: Export des Reports *Artikel* in das RTF-Format

```
Sub ExportAlsWord
    DoCmd.OutputTo acOutputReport, "Artikel", acFormatRTF, "c:\Test.rtf", True
End Sub
```

Neben Objekttyp, Objektname, Format und Dateiname (optional; alternativ fragt Access nach dem Dateinamen) können Sie auch gleich die verknüpfte Anwendung starten lassen (in diesem Fall Microsoft Word).

5.6 Das Printer-Objekt

Auch Access-Programmierer können auf das *Printer*-Objekt bzw. die *Printers*-Collection zugreifen. Doch da gibt es einige Einschränkungen:

- Das *Printer*-Objekt dient nur zur Konfiguration bzw. zur Abfrage der Drucker-Einstellungen
- Das *Printer*-Objekt stellt keinen DC (Gerätekontext) zur Verfügung, damit sind alle Wege zum direkten API-Zugriff verbaut

- Das *Printer*-Objekt verfügt über keinerlei Methoden, diese sind dem übergeordneten Objekt (Bericht) zugeordnet

Lassen Sie sich aber von diesen Einschränkungen nicht beeindrucken. Allein die Möglichkeiten zur Druckerauswahl und Konfiguration sind schon mehr als nur einen Blick wert.

5.6.1 Wo finde ich das Printer-Objekt?

Auf ein *Printer*-Objekt können Sie über

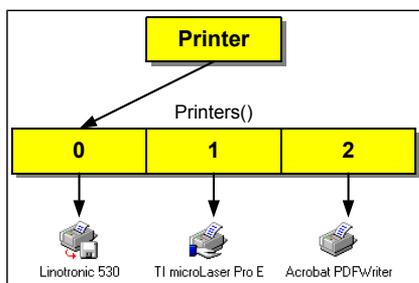
- das *Application*-Objekt,
- die *Application.Printers*-Collection,
- das *Report*-Objekt oder
- das *Form*-Objekt

zugreifen. In allen Fällen handelt es sich zunächst um den aktuell als Standard eingerichteten Systemdrucker. Ändern Sie die Einstellungen des Druckers über *Application.Printer*, ändern sich die Druckereinstellungen für die gesamte Anwendung. Anders bei *Report.Printer* und *Form.Printer*. Hier ändern sich nur die Optionen für das gerade ausgewählte Objekt.

HINWEIS: Ist auf Ihrem System kein Drucker installiert bzw. eingerichtet, kommt es zum Laufzeitfehler 2205, sobald Sie auf eines der *Printer*-Objekte zugreifen. Berücksichtigen Sie dies mit einer entsprechenden Fehlerbehandlung.

5.6.2 Die Printers-Collection

Wie bereits erwähnt, stellt das *Application*-Objekt Ihnen eine Liste (Collection) aller verfügbarer Drucker zur Verfügung. Diese *Printers*-Collection können Sie zum einen zur Anzeige der jeweiligen Druckereigenschaften nutzen, zum anderen besteht auch die Möglichkeit, einen neuen Drucker auszuwählen.



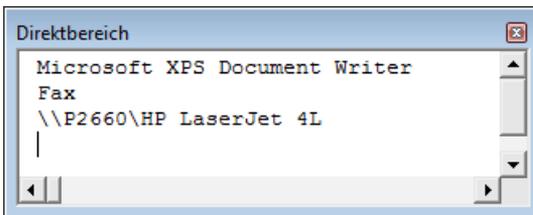
Einige Beispiele werden für Klarheit sorgen.

BEISPIEL: Anzeige des aktuellen Druckers im Direktfenster

```
Sub Test()
    Debug.Print Application.Printer.DeviceName
End Sub
```

BEISPIEL: Anzeige aller installierten Drucker im Direktfenster

```
Sub AllPrinters()
    Dim prn As Printer
    For Each prn In Application.Printers
        Debug.Print prn.DeviceName
    Next
End Sub
```



Auf die einzelnen Elemente der *Printers*-Collection können Sie über die folgenden Varianten zugreifen:

```
Printers!<Gerätename>
Printers("Gerätename")
Printers(Index)
```

HINWEIS: Beachten Sie, dass die Groß-/Kleinschreibung des Gerätenamens von Bedeutung ist. Greifen Sie über den Index auf einen Drucker zu, beginnt die Liste bei 0 (null).

5.6.3 Auswahl eines Druckers

Möchten Sie einen neuen Drucker auswählen, genügt es, wenn Sie dem jeweiligen *Printer*-Objekt ein neues Element aus der *Printers*-Collection zuweisen.

BEISPIEL: Zuweisen eines neuen Druckers

```
Set Application.Printer = Application.Printers(2)
```

Allerdings werden Sie wohl in den seltensten Fällen von dieser simplen Methode Gebrauch machen können. Meist ist es so, dass Sie dem Nutzer die Auswahl des Druckers überlassen wollen/müssen.

BEISPIEL: Auswahl eines Druckers über eine Dialogbox

Mit dem Laden des Formulars füllen wir ein Kombinationsfeld mit den Namen der installierten Drucker:

```
Private Sub Form_Load()
    Dim prn As Printer
    For Each prn In Application.Printers
        Kombinationsfeld1.AddItem prn.DeviceName
    Next
End Sub
```

Der aktuell gewählte Drucker wird im Kombinationsfeld angezeigt:

```
Kombinationsfeld1.Value = Application.Printer.DeviceName
End Sub
```

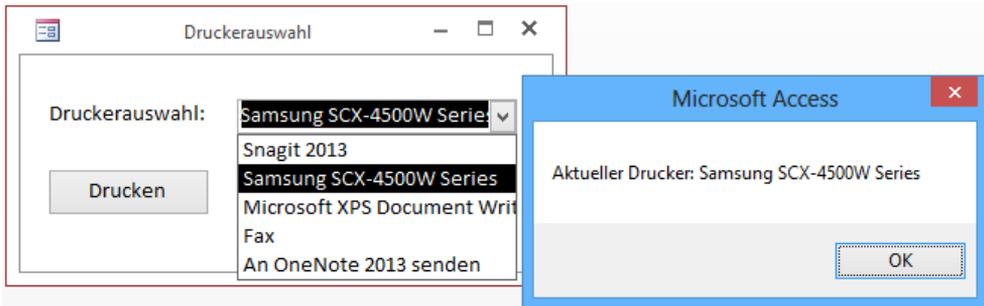
Wählt der Nutzer einen Drucker aus, überschreiben wir das *Printer*-Objekt mit dem neuen Objekt. Gleichzeitig zeigen wir ein Dialogfeld mit dem neuen Namen an:

```
Private Sub Kombinationsfeld1_Change()
    Set Application.Printer = Application.Printers(Kombinationsfeld1.Value)
    MsgBox "Aktueller Drucker: " & Application.Printer.DeviceName
End Sub
```

Der Klick auf die Schaltfläche *Drucken* gibt den Bericht *Bericht1* auf dem gewählten Drucker aus:

```
Private Sub Button1_Click()
    DoCmd.Close acForm, Me.Name
    DoCmd.OpenReport "Bericht1"
End Sub
```

Das zugehörige Formular zeigt die folgende Abbildung:



HINWEIS: Beachten Sie, dass der Drucker vor dem Aufruf eines Berichts ausgewählt werden muss. Weiterhin dürfen *Printer*-Eigenschaften nicht aus dem Bericht heraus geändert werden (Ereignismethoden), sondern nur über ein aufrufendes Formular bzw. eine aufrufende Prozedur.

BEISPIEL: Die folgende Anweisung ist nicht zulässig und führt zu einem Laufzeitfehler.

```
Private Sub Report_Open(Cancel As Integer)
    Set Me.Printer = Application.Printers(2) ' -> Fehler
...
End Sub
```

HINWEIS: Die Einstellungen über *Application.Printer* gehen mit dem Beenden der Access-Anwendung verloren, d.h., Sie müssen diese eventuell in der Registry sichern und beim Neustart wiederherstellen.

5.6.4 Speichern von Berichts-Optionen

Im Gegensatz zum *Application.Printer*-Objekt ist es bei Formularen und Berichten möglich, die geänderten Druck-Einstellungen zu sichern. Beim erneuten Aufruf stehen die alten Optionen automatisch wieder zur Verfügung.

Zwei Weg bieten sich an:

- Einstellen der Optionen über den Dialog *Seite einrichten*
- Ändern der *Report.Printer*-Eigenschaften und Sichern des Berichts

Die zweite Variante ist mit relativ geringem Aufwand aus dem Quellcode heraus realisierbar.

BEISPIEL: Ändern der Seitenausrichtung auf "Querformat" und Auswahl des Fax-Druckers

```
Dim rep As Report

DoCmd.OpenReport "Report1", acViewDesign, , , acHidden
Set rep = Reports("Report1")
rep.Printer = Application.Printers(2)
rep.Printer.orientation = acPRORLandscape
DoCmd.Close acReport, "Report1", acSaveYes ' Sichern der Optionen
```

Ändern Sie mit einer der beschriebenen Methoden die Druck-Optionen, legt Access automatisch eine spezielle Datenstruktur an, in der die *Printer*-Eigenschaften gesichert werden.

Sie können im späteren Programmverlauf über die Abfrage der *UseDefaultPrinter*-Eigenschaft (*True/False*) testen, ob spezielle Optionen für ein Bericht/Formular vorhanden sind.

BEISPIEL: Ändern der Report-Einstellungen, wenn noch keine Optionen festgelegt wurden

```
Dim rep As Report
DoCmd.OpenReport "Report1", acViewDesign, , , acHidden
Set rep = Reports("Report1")
If rep.UseDefaultPrinter Then
    rep.Printer = Application.Printers(2)
    ...
```

Damit dürften Sie die wichtigsten Möglichkeiten in Bezug auf das *Printer*-Objekt in Access kennen gelernt haben. Was bleibt, ist eine Zusammenfassung der *Printer*-Eigenschaften.

5.6.5 Eigenschaften des Printers

Die folgende Tabelle zeigt die wichtigsten Eigenschaften und deren Bedeutung.

Eigenschaft	Beschreibung/Hinweise
<i>ColorMode</i>	<i>acPRCMMonochrome</i> = monochrome Druckausgabe (Graustufen), <i>acPRCMColor</i> = farbige Druckausgabe
<i>Copies</i>	Die Anzahl der Kopien (Lesen/Schreiben)
<i>DeviceName</i> , <i>DriverName</i> , <i>Port</i>	... Informationen über den aktuellen Drucker Beispiel: Linotronic, winspool, LPT1
<i>Duplex</i>	... legt fest, ob und wie eine Seite beidseitig bedruckt wird
<i>Orientation</i>	Ausrichtung der Seite (<i>acPRORLandscape</i> = Querformat, <i>acPRORPortrait</i> = Hochformat)
<i>PaperBin</i>	Bestimmt, woher das Papier für den Druck kommt: <i>acPRBNAuto</i> , <i>acPRBNCassette</i> , <i>acPRBNEvelope</i> , <i>acPRBNEnvManual</i> , <i>acPRBNFormSource</i> , <i>acPRBNLargeCapacity</i> , <i>acPRBNLargeFmt</i> , <i>acPRBNLower</i> , <i>acPRBNManual</i> , <i>acPRBNMiddle</i> , <i>acPRBNSmallFmt</i> , <i>acPRBNTractor</i> , <i>acPRBNUpper</i>
<i>PaperSize</i>	Die Papier- bzw. Seitengröße, z.B. <i>acPRPSA3</i> = DIN A3, 297 x 420 mm <i>acPRPSA4</i> = DIN A4, 210 x 297 mm <i>acPRPSA4Small</i> = DIN A4 klein, 210 x 297 mm <i>acPRPSA5</i> = DIN A5, 148 x 210 mm
<i>PrintQuality</i>	Die Druckqualität: <i>acPRPQDraft</i> = Entwurfsauflösung <i>acPRPQLow</i> = Geringe Auflösung <i>acPRPQMedium</i> = Mittlere Auflösung <i>acPRPQHigh</i> = Hohe Auflösung Der Rückgabewert ist die Auflösung in DPI (Dots Per Inch)

Neben den oben aufgeführten Eigenschaften, die Sie zum Beispiel über das *Application.Printer*-Objekt abrufen und setzen können, gibt es auch eine Reihe von Eigenschaften, die Report-spezifisch sind:

Eigenschaft	Beschreibung
<i>BottomMargin</i>	Der untere Seitenrand in Twips (567 Twips = 1 cm)
<i>ColumnSpacing</i>	Spaltenabstand (Fehler in der Online-Hilfe)
<i>DataOnly</i>	Ist dieser Wert <i>True</i> , werden nur die eigentlichen Daten, nicht die Oberflächenelemente der Datenblattansicht gedruckt
<i>DefaultSize</i> , <i>ItemSize-Height</i> , <i>ItemSizeWidth</i>	Ist dieser Wert <i>False</i> , werden die Werte <i>ItemSizeHeight</i> und <i>ItemSizeWidth</i> für die Bestimmung der Größe des Detailabschnitts herangezogen, andernfalls werden die Werte ignoriert
<i>ItemsAcross</i>	Die Anzahl der Spalten pro Seite
<i>ItemLayout</i>	Legt die Druckrichtung bei mehrspaltigen Reports fest
<i>LeftMargin</i>	Der linke Seitenrand in Twips

Eigenschaft	Beschreibung
<i>RightMargin</i>	Der rechte Seitenrand in Twips
<i>RowSpacing</i>	Zeilenabstand (Fehler in der Online-Hilfe)
<i>TopMargin</i>	Der obere Seitenrand in Twips

5.7 Direkte Druckausgabe

Erinnern Sie sich noch an die Möglichkeiten der Druckerausgabe unter DOS? Wenn ja, dann dürften Ihnen die folgenden Zeilen recht bekannt vorkommen.

```
Open "LPT1" For output As #1
Print #1 ,"bla bla ..."      ' Ausgabe Text
Print #1 , Chr(12)             ' neue Seite
Close #1                       ' Ausgabe schließen
```

Die gleiche Vorgehensweise ist auch unter Windows bzw. Access möglich, sollte jedoch weitgehend vermieden werden. Zum einen "verschenken" Sie die Möglichkeiten, die der jeweilige Druckertreiber bietet, zum anderen machen Sie Ihr Programm wieder hardwareabhängig. Dies insbesondere wenn es darum geht, bestimmte Zeichensätze auszuwählen oder bestimmte Schriftattribute (Kursiv, Fett etc.) zu aktivieren. Zwangsläufig werden Sie sich wieder mit den bekannten (und berechtigten) ESCAPE-Sequenzen beschäftigen müssen. Die Folge: Ihr Programm läuft mit dem Drucker des Herstellers A, aber nicht mehr mit einem Drucker des Herstellers B. Warum dann dieser Aufwand? Die Antwort findet sich wie so oft in den praktischen Gegebenheiten. Beispielsweise werden noch vielfach sehr spezielle Formular-Drucker eingesetzt, für die es keine sinnvollen Windows-Treiber gibt.

5.8 Übersichten

5.8.1 DrawMode-Eigenschaft

Code	Erläuterung
1	Schwarzer Stift
2	Inverse Einstellung 15 (<i>NotMergePen</i>)
3	Kombination der Hintergrundfarbe mit der inversen Stiftfarbe (<i>MaskNotPen</i>)
4	Inverse Einstellung 13 (<i>NotCopyPen</i>)
5	Kombination der Stift- mit der inversen Hintergrundfarbe (<i>MaskPenNot</i>)
6	Invertieren der Hintergrundfarbe (<i>Invert</i>)
7	XOR-Kombination der Stift- und der Hintergrundfarbe (<i>XorPen</i>)
8	Inverse Einstellung zu 9 (<i>NotMaskPen</i>)
9	Kombination der Stift- und der Hintergrundfarbe (<i>MaskPen</i>)
10	Inverse Einstellung zu 7 (<i>NotXorPen</i>)

Code	Erläuterung
11	Keine Operation (<i>NOP</i>)
12	Kombination der Hintergrund- mit der inversen Stiftfarbe (<i>MergeNotPen</i>)
13	Voreinstellung, die Stiftfarbe entspricht der ForeColor-Property (<i>CopyPen</i>)
14	Kombination der Stiftfarbe und der inversen Hintergrundfarbe (<i>MergePenNot</i>)
15	Kombination der Stiftfarbe und der Hintergrundfarbe (<i>MergePen</i>)
16	Weißer Stift

5.8.2 Farbkonstanten

Konstante	Wert	Beschreibung
<i>vbBlack</i>	&H0	Schwarz
<i>vbRed</i>	&HFF	Rot
<i>vbGreen</i>	&HFF00	Grün
<i>vbYellow</i>	&HFFFF	Gelb
<i>vbBlue</i>	&HFF0000	Blau
<i>vbMagenta</i>	&HFF00FF	Magenta
<i>vbCyan</i>	&HFFFFFF00	Zyan
<i>vbWhite</i>	&HFFFFFFF	Weiß

5.9 Praxisbeispiele

5.9.1 Aufruf eines Berichts mit Datenfilter

Report-Objekt: *FilterOn*-Eigenschaft, Berichts-Ansicht, Layout-Ansicht, Zeilen nummerieren;
DoCmd-Objekt: *OpenReport*-Methode; *CDate*- und *Format*-Funktion; *Select Case*-Anweisung;
OptionGroup-Steuerelement; *Application*-Objekt: *TempVars*-Auflistung;

In vielen Fällen erfolgt der Aufruf eines Berichts nicht direkt, sondern aus einem vorgeschalteten Formular heraus (über Menüleiste oder Schaltfläche). Dabei werden im Allgemeinen die für die Filterung der anzuzeigenden Datensätze notwendigen Parameter vorher interaktiv im Formular festgelegt.

Für den Aufruf des Berichts über die *OpenReport*-Methode des *DoCmd*-Objekts gibt es verschiedene Möglichkeiten:

- *DoCmd.OpenReport Berichtsname, Ansicht, Filtername, Bedingung*
- *DoCmd.OpenReport Berichtsname, Ansicht, Filtername*
- *DoCmd.OpenReport Berichtsname, Ansicht*

Wir wollen uns hier auf die erste Variante beschränken. Als Beispiel stellen wir uns die Aufgabe, das Anlagevermögen der Tabelle *BGA* unserer Beispieldatenbank für ein bestimmtes Zeitfenster aufzulisten.

Dazu brauchen wir zwei neue Access-Objekte: Ein Formular und einen Bericht.

Oberfläche des Formulars

Da es sich um ein reines Bedienformular handelt, bleibt die Eigenschaft *Datensatzquelle* unbesetzt. Verändern Sie aber die Eigenschaften *Datensatzmarkierer* und *Navigationsschaltflächen* in *Nein*.

Beide Textfelder sind ungebunden. Rechts befindet sich eine ungebundene *Optionsgruppe* mit fünf *Optionsfeldern* zur Auswahl der gewünschten Ansicht.

Quelltext für Formular

Zu Beginn werden standardmäßige Datumswerte eingestellt:

```
Private BeginnDatum As Date, EndeDatum As Date
Private Filter1 As String

Private Sub Form_Load()
    ' Start
    BeginnDatum = Date - 365 ' vor etwa einem Jahr
    EndeDatum = Date ' heute
    Text1.Value = Format$(BeginnDatum, "d.m.yyyy")
    Text2.Value = Format$(EndeDatum, "d.m.yyyy")
End Sub
```

Die Datumswerte sollen in der Report-Überschrift erscheinen und werden deshalb in der folgenden Prozedur als globale Variablen gespeichert. Hauptaufgabe aber ist die Definition des Filters, wobei die Datumsvariablen in "englische" Strings umformatiert werden müssen (siehe SQL-Kapitel 15):

```
Private Sub setFilter1() ' setzt globale Variablen und Filter
    TempVars![Datum1] = Text1.Value
    TempVars![Datum2] = Text2.Value
```

```

BeginnDatum = CDate(Text1.Value)
EndeDatum = CDate(Text2.Value)
Filter1 = "RDatum BETWEEN " & Format$(BeginnDatum, "\#m\d\yyyy\#") & _
         " AND " & Format$(EndeDatum, "\#m\d\yyyy\#")
End Sub

```

Der Bericht wird nun durch Übergabe entsprechender *AcView*-Konstanten (siehe Seite 263) mit der gewünschten Ansicht aufgerufen:

```

Private Sub Befehl1_Click()      ' Bericht aufrufen
    Call setFilter1
    Select Case Rahmen1.Value
        Case 1: DoCmd.OpenReport "Bericht1", acViewPreview, , Filter1 ' Seiten-Ansicht
        Case 2: DoCmd.OpenReport "Bericht1", acViewReport, , Filter1 ' Report-Ansicht
        Case 3: DoCmd.OpenReport "Bericht1", acViewDesign, , Filter1 ' Entwurfs-Ansicht
        Case 4: DoCmd.OpenReport "Bericht1", acViewLayout, , Filter1 ' Layout-Ansicht
        Case 5: DoCmd.OpenReport "Bericht1", acViewNormal, , Filter1 ' Drucken
    End Select
End Sub

```

Oberfläche des Berichts

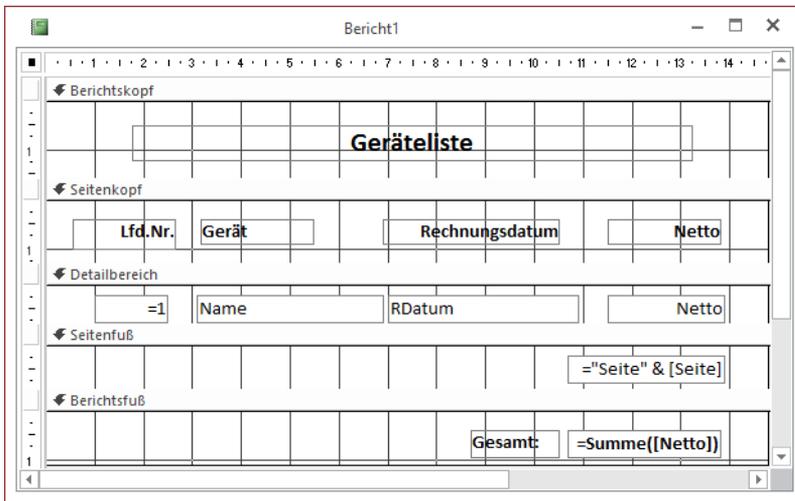
Öffnen Sie einen neuen Bericht in der Entwurfsansicht. Ändern Sie gleich zu Beginn die Standard-eigenschaften des Berichts entsprechend der folgenden Abbildung:



Sind Sie mit dem Berichtsgenerator einigermaßen vertraut, dürfte es für Sie keine Hürde sein, die in der folgenden Abbildung gezeigte Oberfläche "zusammenzuschieben".

Beachten Sie folgende Besonderheiten:

- Die Überschrift ist in einem *Bezeichnungsfeld* untergebracht und wird durch Setzen der *Caption*-Eigenschaft erst zur Laufzeit mit den Datumsangaben komplettiert.
- Um die Datensätze in der *LfdNr*-Spalte durchzunummerieren, legen Sie die Eigenschaft *Steuerelementinhalt* des Textfelds auf "=1" fest und die Eigenschaft *Laufende Summe* auf "Über Alles".



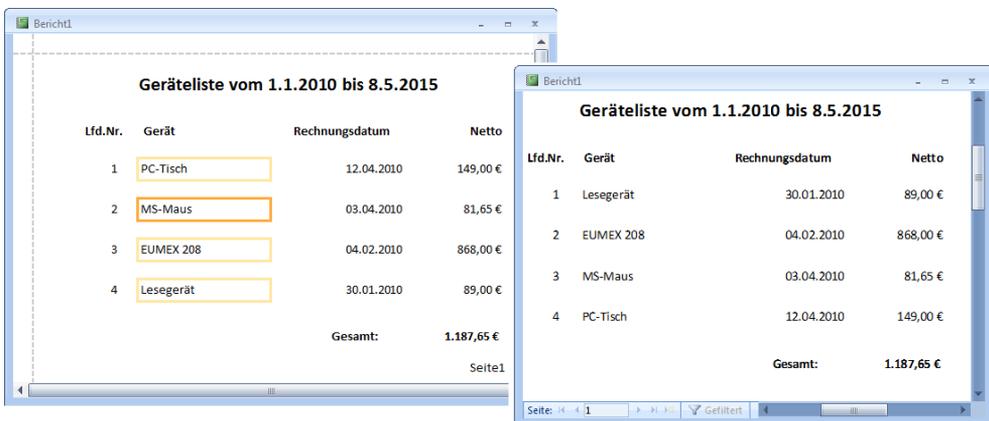
Quelltext des Berichts

Beim Öffnen des Berichts werden die Seitenüberschrift angepasst und das Filter eingeschaltet:

```
Private Sub Report_Open(Cancel As Integer)
    Bezeichnungsfeld0.Caption = "Geräteleiste vom " & Format$(TempVars![Datum1], "d.m.yyyy") & _
        " bis " & Format$(TempVars![Datum2], "d.m.yyyy")
    Me.FilterOn = True
End Sub
```

Test

Öffnen Sie das Formular, geben Sie ein Anfangs- und ein Enddatum ein und überzeugen Sie sich von den Besonderheiten der verschiedenen Report-Ansichten.



Bemerkungen

- Die Layout-Ansicht erlaubt es Ihnen, auch zur Laufzeit am Erscheinungsbild des Berichts herumzudoktern (Steuerelemente verschieben etc.). Dazu muss die Eigenschaft *AllowLayoutView* ("Layoutansicht zulassen") des Reports *True* sein (Standard).
- Die Übergabe von Datumsangaben etc. an den Report ist auch über den *OpenArgs*-Parameter von *DoCmd.OpenReport* möglich (siehe Seite 263), sodass auf globale Variable (wie im vorliegenden Beispiel) verzichtet werden kann.
- Wenn Sie wollen, können Sie im *OpenReport*-Ereignis auch noch bestimmte Formatierungen setzen, z.B. eine alternierende Zeilenfarbe einstellen (siehe *Section*-Objekt Seite 271):

```
Me.Section(acDetail).AlternateBackColor = vbYellow
```

5.9.2 Im Report gruppieren und rechnen

Report-Objekt: *Format*-Ereignis; Funktionszuweisungen in *ControlSource*-Eigenschaft; *Format*-Funktion; *Date*-Typ; Gruppieren (Report);

Für das Finanzamt soll eine Übersicht über das Anlagevermögen (Tabelle *BGA* der Beispieldatenbank dieses Kapitels) erstellt werden und zwar getrennt für Anschaffungen unter- und oberhalb der 400-Euro-Grenze (bezogen auf den Netto-Anschaffungspreis).

Die hier vorgestellte Lösung zeigt, wie man die Fähigkeiten des Berichtsgenerators sinnvoll durch VBA-Code ergänzen kann.

Oberfläche

Öffnen Sie einen neuen Bericht, dessen *Datensatzquelle* auf *BGA* zu setzen ist. Aktivieren Sie über das Kontextmenü die Ansicht *Berichtskopf/-fuß*.

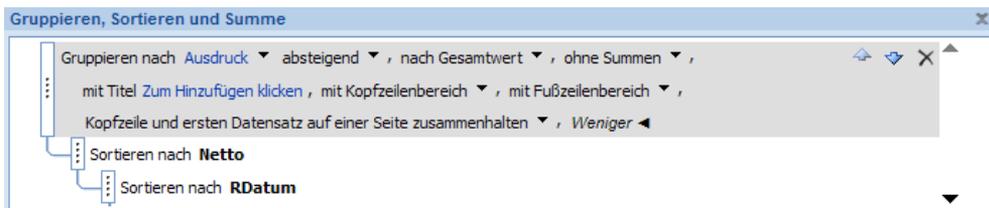
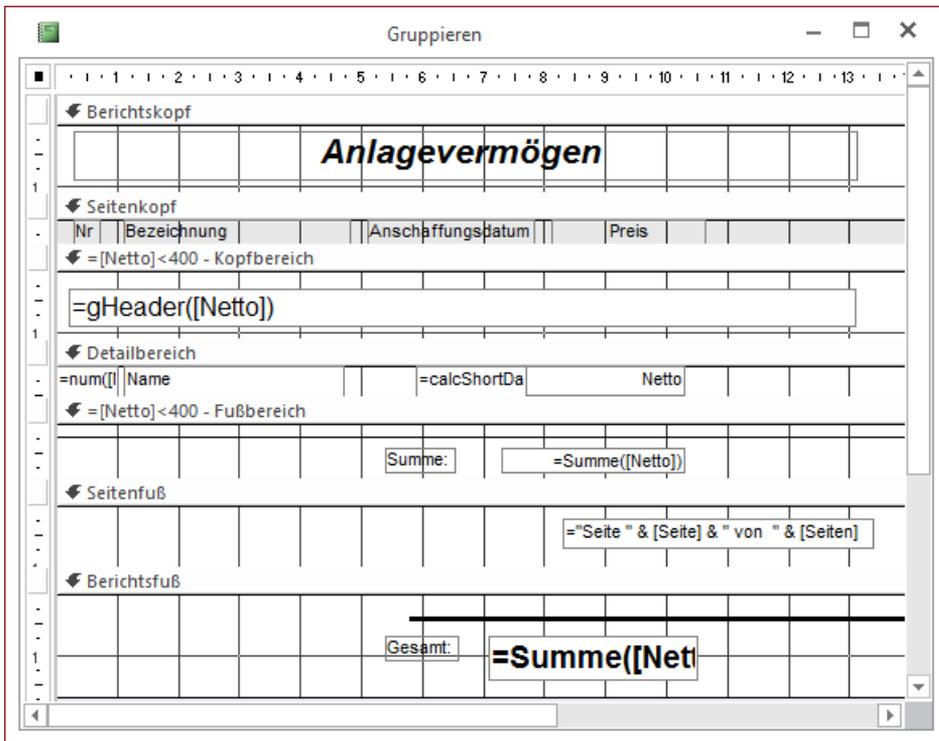
Bestücken Sie die Oberfläche wie folgt mit *Bezeichnungs-* und *Textfeldern* (von oben nach unten, von links nach rechts):

Berichtskopf Ein Bezeichnungsfeld (*Bezeichnungsfeld0*)

Seitenkopf Vier Bezeichnungsfelder (*Bezeichnungsfeld1* bis *Bezeichnungsfeld4*)

Detailbereich Vier Textfelder (*Text1* bis *Text4*), *Text2* und *Text4* werden über ihre *Steuerelementinhalt*-Eigenschaft direkt mit den Feldern *Name* und *Netto* verbunden.

Gruppenkopf Um, wie abgebildet, einen Gruppenbereich für das Kriterium "*Netto < 400*" einzufügen, wählen Sie den Befehl *Gruppieren und sortieren* (Registerkarte *Gruppierung und Summen*) und stellen im entsprechenden Dialog die Sortierkriterien gemäß der Abbildung ein.



Innerhalb der Gruppe wird nach fallenden Nettobeträgen geordnet, bei Gleichheit wird das Rechnungsdatum herangezogen. Die variable Zwischenüberschrift erscheint in einem später anzubindenden Textfeld (*Text5*).

Gruppenfuß Ein vorerst ungebundenes Textfeld (*Text6*) dient der Anzeige der jeweiligen Gruppensumme.

Seitenfuß Am unteren Rand rechts soll die Seitenziffer (und die Gesamtanzahl der Seiten) in einem Textfeld (*Text7*) bereitgestellt werden. Den Steuerelementinhalt weisen wir am besten mit dem Ausdrucks-Generator zu (*Gebäuchliche Ausdrücke, Seite N von M, Schaltfläche Einfügen*).

Berichtsfuß Hier soll in einem, vorerst ungebundenen, Textfeld (*Text8*) die Gesamtsumme erscheinen.

Bereits in diesem Zwischenstadium können Sie den Bericht erstmalig starten. Name und Nettopreis aller Geräte müssen, in zwei Gruppen aufgeteilt, aufgelistet werden. Vergessen Sie aber nicht, die *Format*-Eigenschaft der Textfelder (*Text4* bis *Text9*) auf *Währung* bzw. *Euro* umzustellen.

Quelltext

```

Private nrg, nrb As Integer                ' Zähler für lfd. Nr

Private Function calcShortDate(RDatum As Date) As String    ' Kurzform Monat/Jahr aus Datum
    ermitteln
    calcShortDate = Format$(RDatum, "mm\yy")
End Function

Private Function num(netto As Currency) As String          ' zählt BGA u. GWG getrennt
    If netto >= 400 Then
        nrg = nrg + 1
        num = Str(nrg) & "."                ' BGA
    Else
        nrb = nrb + 1
        num = Str(nrb) & "."                ' GWG
    End If
End Function

Private Function gHeader(netto As Currency) As String      ' setzt Gruppenüberschrift
    If netto < 400 Then
        gHeader = "II. Geringwertige Gebrauchsgüter (GWG)"
    Else
        gHeader = "I. Betriebsgrundausrüstungen (BGA)"
    End If
End Function

Private Sub Gruppenkopf0_Format(Cancel As Integer, FormatCount As Integer)
    nrg = 0                                ' Zähler bei Gruppenanfang initialisieren
    nrb = 0
End Sub

Private Sub Report_Open(Cancel As Integer)                ' Start
' Zuweisen der Steuerelementeinhalte:
Text1.ControlSource = "=num([Netto])"                ' lfd. Nr. innerhalb Gruppe
Text3.ControlSource = "=calcShortDate([RDatum])"
Text5.ControlSource = "=gHeader([Netto])"            ' Zwischenüberschrift für Gruppen
Text6.ControlSource = "=Sum([Netto])"
Text8.ControlSource = Text6.ControlSource            ' Endsumme
End Sub

```

Test

Öffnen Sie den Bericht und überzeugen Sie sich von seiner Funktionsfähigkeit:

Im Bericht rechnen und gruppieren

Anlagevermögen

Nr	Bezeichnung	Anschaffungsdatum	Preis
I. Betriebsgrundausrüstungen (BGA)			
15.	PC-Pentium 90	08/03	4.729,00 €
16.	Transceiver FT 890	02/03	2.345,00 €
17.	32MB-RAM-Aufrüstung	03/03	1.750,00 €
18.	17" Monitor	10/03	1.499,00 €
19.	Laserdrucker	08/03	1.399,00 €
20.	SVGA-ATI-Grafikkarte	10/04	899,00 €
21.	EUMEX 208	02/07	868,00 €
20.	PC 386 SX25	10/05	750,00 €
21.	Testgerät	12/07	736,00 €
22.	Kopierer Z20	07/06	702,77 €
23.	KW-Groundplane Antenne	02/04	654,53 €
24.	15" Monitor	09/06	649,00 €
25.	Kopiergerät	06/07	620,00 €
26.	PACTOR-KW-Modem	09/04	455,00 €
27.	PEARL-Telefonmodem	02/05	444,70 €
Summe:			18.501,00 €
II. Geringwertige Gebrauchsgüter (GWG)			
29.	Funkmodem TNC 2C	02/04	383,69 €
30.	Netzteil DM 120	01/04	355,00 €
31.	4 MB RAM-Modul	08/05	350,00 €
32.	UKW-Handfunke (1)	07/06	339,00 €
33.	Digital Diary CASIO SF-800C	05/07	320,00 €
34.	CD-ROM-Laufwerk	10/06	299,00 €
35.	IOMEGA-Streamer	03/05	299,00 €

Nr	Bezeichnung	Anschaffungsdatum	Preis
23.	Tastatur	11/07	25,22 €
24.	Windows 95 Tastatur	11/07	25,22 €
25.	PC-Verbindungskabel	09/06	19,90 €
26.	MS-Maus	04/07	8,65 €
Summe:			4.207,16 €
Gesamt:			22.708,16 €

Bemerkungen

- Natürlich hätten wir statt der direkten Zuweisung der *ControlSource*-Eigenschaft auch die Eigenschaft *Steuerelementinhalt* für die noch ungebundenen Textfelder direkt in der Entwurfsansicht des Formulars setzen können. Allerdings sind dabei Besonderheiten in der Schreibweise zu beachten. Statt der Kommas sind Semikolons als Trennzeichen innerhalb der Liste der Übergabeparameter zu verwenden, und für die Aggregatfunktionen sind die deutschen Bezeichner einzusetzen.

BEISPIEL: Statt des VBA-Codes

```
Text7.ControlSource ="=Sum(calcafa([RDatum],[Netto],[Afa]))"
```

können Sie auch den *Steuerelementinhalt* von *Text7* zur Entwurfszeit mittels Ausdrucks-Editor zuweisen:

```
=Summe(calcafa([RDatum];[Netto];[Afa]))
```

- Die *CreateGroupLevel*-Methode ist leider nur zur Entwurfszeit verfügbar und damit lediglich zur Unterstützung der Entwicklung eigener Berichtsassistenten sinnvoll. Ebenfalls mit Einschränkungen verwendbar sind die gruppenspezifischen Eigenschaften *SortOrder*, *GroupOn*, *GroupInterval*, *KeepTogether* etc.

5.9.3 Erstellen und Drucken eines Diagramms

Diagramm; SQL: Abfrage, SUM, INNER JOIN, GROUP BY

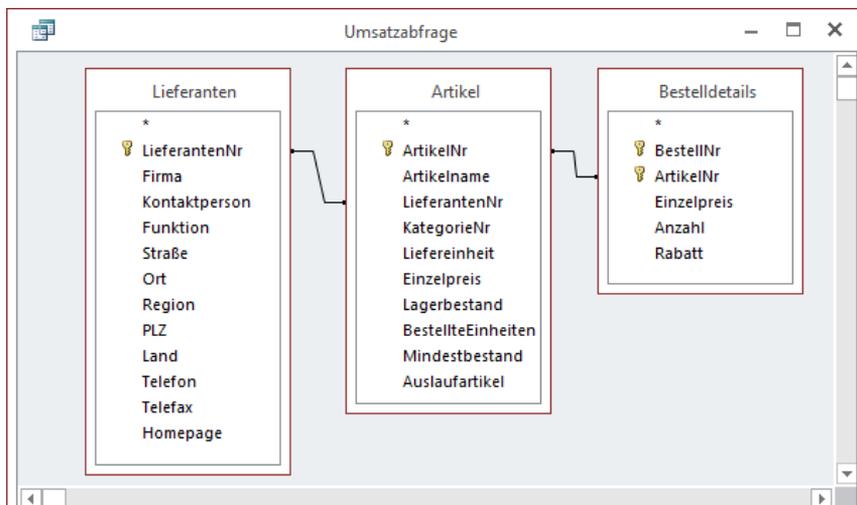
Schluss mit langweiligen Listen! In diesem Beispiel entwerfen wir ein Diagramm und fügen dieses als Unterbericht in unseren Report ein.

HINWEIS: Seit Access 2013 wird das Pivot-Diagramm nicht mehr unterstützt. Sie müssen mit dem *Diagramm*-Control oder Microsoft Excel arbeiten.

Entwurf Abfrage

Bevor Sie sich gleich an den Diagrammentwurf machen, sollten Sie zunächst die Daten entsprechend aufbereiten, sodass 1:n-Beziehungen auch mit den gewünschten Detaildatensätzen dargestellt werden können. Dazu verwenden Sie am besten eine Access-Abfrage.

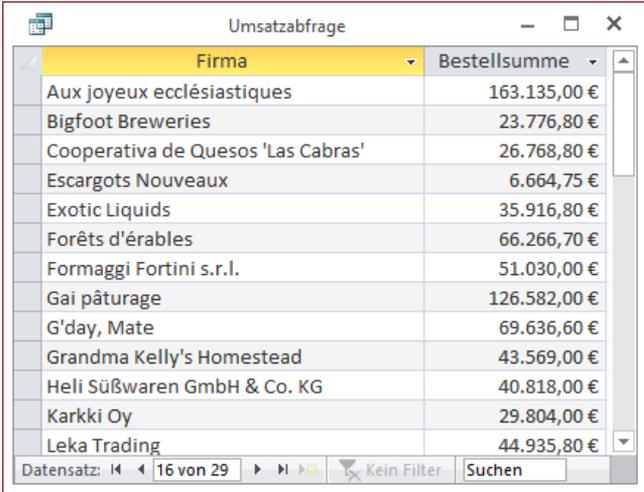
Für unser Beispiel möchten wir die erzielten Umsätze der einzelnen Lieferfirmen berechnen. Dazu benötigen wir die folgenden Tabellen.



Die entsprechende SQL-Abfrage liefert uns zwei Spalten und für jede Firma die Summe der Umsätze:

```
SELECT
  Lieferanten.Firma,
  SUM(Bestelldetails.Einzelpreis*Bestelldetails.Anzahl) AS Bestellsumme
FROM
  Lieferanten INNER JOIN
  (Artikel INNER JOIN Bestelldetails ON Artikel.ArtikelNr =
  Bestelldetails.ArtikelNr) ON
  Lieferanten.LieferantenNr = Artikel.LieferantenNr
GROUP BY
  Lieferanten.Firma;
```

Das Ergebnis der Abfrage zeigt die folgende Abbildung:



Firma	Bestellsumme
Aux joyeux ecclésiastiques	163.135,00 €
Bigfoot Breweries	23.776,80 €
Cooperativa de Quesos 'Las Cabras'	26.768,80 €
Escargots Nouveaux	6.664,75 €
Exotic Liquids	35.916,80 €
Forêts d'érables	66.266,70 €
Formaggi Fortini s.r.l.	51.030,00 €
Gai pâturage	126.582,00 €
G'day, Mate	69.636,60 €
Grandma Kelly's Homestead	43.569,00 €
Heli Süßwaren GmbH & Co. KG	40.818,00 €
Karkki Oy	29.804,00 €
Leka Trading	44.935,80 €

Speichern Sie die Abfrage unter dem Namen *Umsatzabfrage* ab.

Erstellen Bericht

Markieren Sie die neue Abfrage *Umsatzabfrage* und wählen Sie im Menüband die Funktion *ERSTELLEN/Berichtsentwurf*. Es öffnet sich der Berichtsdesigner.

Verkleinern Sie den Detailbereich des neuen Berichts und fügen Sie in den Kopfbereich ein Diagramm ein. Der daraufhin eingblendete Assistent kümmert sich um die nächsten Schritte und nimmt uns zunächst etwas Arbeit ab.

Diagramm-Assistent

Welche Tabelle oder Abfrage möchten Sie zur Erstellung Ihres Diagramms verwenden?

Abfrage: Umsatzabfrage

Ansicht

Tabellen Abfragen Beides

Abbrechen < Zurück Weiter > Fertig stellen

Wir benötigen beide Felder der Abfrage (Zeilen und Spalten):

Diagramm-Assistent

Welche Felder enthalten die Daten für das Diagramm?

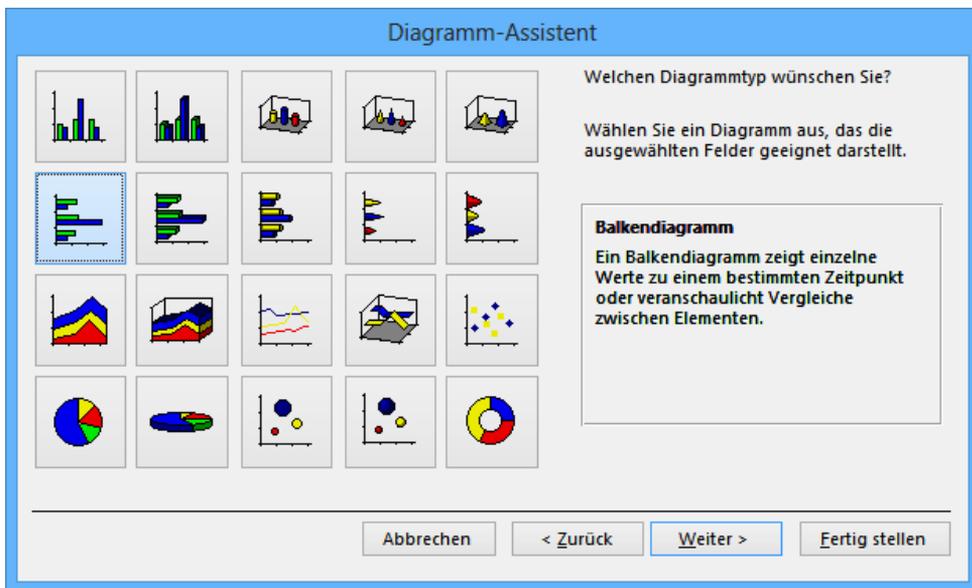
Verfügbare Felder:

Felder für das Diagramm:

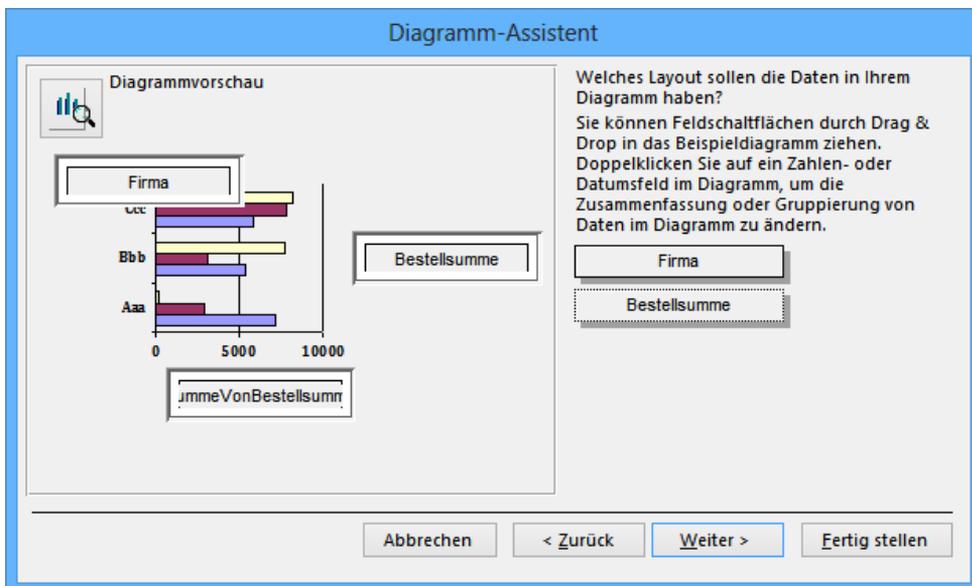
Firma
Bestellsumme

Abbrechen < Zurück Weiter > Fertig stellen

Nachfolgend müssen Sie sich für einen Diagrammtyp entscheiden. Dabei sollten Sie die Anzahl der Zeilen/Spalten berücksichtigen, damit diese auch sinnvoll dargestellt werden können.



Der Assistent hat bereits die richtige Zuordnung von x- und y-Achse getroffen:

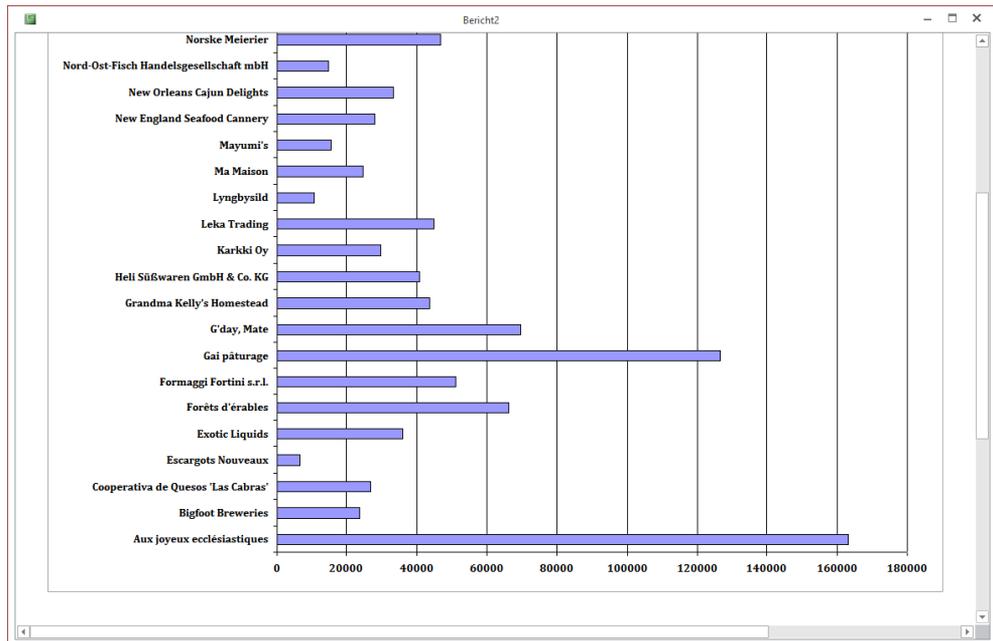


Damit können wir den Assistenten beenden und das Diagramm optisch anpassen.

Mit einem Doppelklick öffnen Sie das *Diagramm*-Objekt und können jetzt wie in Excel an den Elementen des Diagramm "herumspielen", d.h. Schriftarten und Formatierungen setzen, Elemente ausblenden (Legende) etc.

HINWEIS: Wundern Sie sich nicht, dass zur Entwurfszeit nur ein Platzhalter-Diagramm angezeigt wird, zur Laufzeit erscheint das gewünschte Diagramm.

Passen Sie nun noch das Seitenformat an und schon können Sie die Druckvorschau aufrufen:



5.9.4 Berichte in Formularen anzeigen

Unterbericht; Formular

Sie können auch einen Bericht als Unterbericht in einem Formular anzeigen lassen und somit gleich eine Vorschau auf das spätere Druckergebnis generieren.

Bericht erstellen

Erstellen Sie zunächst per Assistent einen einfachen Bericht für die Tabelle *Artikel* (siehe Beispieldatenbank). Speichern Sie diesen unter dem Namen *Artikel* ab.

Formular erstellen

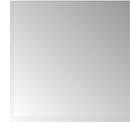
Erzeugen Sie jetzt (ebenfalls per Assistent) ein Formular für die Tabelle *Kategorien*. Ziehen Sie in der Entwurfsansicht den schon erstellten Bericht *Artikel* per Drag & Drop in das Formular, um diesen als Unterbericht einzufügen (siehe Laufzeitanzeige). Verbinden Sie Formular und Unterbericht über die Eigenschaften *Verknüpfen nach* bzw. *Verknüpfen von* des Unterberichts, um jeweils alle Artikel der im Formular ausgewählten Kategorie anzuzeigen.



Test

Nach dem Start sollten Sie, je nach gewählter Kategorie, die passenden Artikel im untergeordneten Bericht sehen:





Index

A

- AbbrechenDatensatzänderung 179
- Abfragen 381
- Abs 116
- AbsolutePage 472
- AbsolutePosition 386, 417, 472
- ACCDR 997
- Access 2003-Menü 1096
- Access-Bibliothek 1019
- Access-Optionen 1035
- AccessControls.xlsx 1044
- AccessObject 344
- acCmdCopy 1104
- acCmdPaste 1104
- acCmdWindowHide 984
- acDetail 203, 316
- acFooter 203, 316
- acFormatPDF 288
- acFormatRTF 288
- acGroupLevelIHeader 316
- acHeader 203, 316
- acOLECreateEmbed 1123
- acOLECreateLink 1123
- acPageFooter 316
- acPageHeader 316
- acQuitPrompt 996
- acSysCmdAccessDir 1024
- acSysCmdAccessVer 1024
- acSysCmdClearStatus 982
- acSysCmdGetWorkgroupFile 1024
- acSysCmdInitMeter 983
- acSysCmdRemoveMeter 983
- acSysCmdRuntime 997, 1024
- acSysCmdSetStatus 982
- acSysCmdUpdateMeter 983
- acTextFormatHTMLRichText 224
- acTextFormatPlain 224
- Action 1123
- Activate 208, 273
- Active Data Objects Extensions 485
- ActiveConnection 449, 472, 476, 489, 621
- ActiveControl 232
- ActiveForm 232
- ActiveForm.hwnd 803
- ActiveReport 232
- ActiveX 233, 1119
- ActiveX Data Objects 437
- ActiveX-Code-Komponente 1127
- ActiveX-Komponente 233
- AcView 262
- AcWindowMode 262
- adAsyncConnect 447
- adAsyncExecute 454
- adAsyncFetch 455
- adBookmark 456
- adBookmarkCurrent 462
- adBookmarkFirst 462
- adBookmarkLast 462
- adCmdStoredProc 450, 454
- adCmdTable 450, 454
- adCmdTableDirect 454
- adCmdText 450, 454
- adCmdUnknown 450
- adColFixed 492
- adColNullable 492
- adCommandFile 454
- Add 327, 357

- ADD COLUMN 962
- Add-In 999
- AddFromFile 331, 333
- AddFromGUID 333
- AddFromString 331
- AddItem 242
- Addition 102
- AddMenu 70
- AddNew 387, 392, 412, 428, 458, 473, 479
- AddSharedImage 257
- adIndexNullsDisallow 494
- adIndexNullsIgnore 494
- adIndexNullsIgnoreAny 494
- adKeyForeign 494
- adKeyPrimary 494
- adKeyUnique 494
- adLockBatchOptimistic 459
- ADO 437
- ADO Query 901
- ADO-Frontend 479
- ADO-Stream laden 866
- ADO-Version 560
- ADODB 335, 440
- ADODB.Error 621
- ADOR 440
- ADOX 440, 485
- ADP-Projekt 565
- adRiCascade 497
- adRiNone 497
- adRiSetDefault 497
- adRiSetNull 497
- adUseClient 446
- adUseServer 446
- adVarChar 490
- AdventureWorks 583
- aEditAdd 457
- aEditDelete 457
- aEditInProgress 457
- aEditNone 457
- AfterBeginTransaction 212
- AfterDelConfirm 212
- AfterDelete 175
- AfterInsert 175, 212
- AfterUpdate 175, 212, 230, 248
- Aggregatfunktionen 944
- Aktionsabfrage 382
- Alias 1113
- Alignment 237
- ALL 922
- AllForms 345
- AllowAdditions 206
- AllowBypassKey 994
- AllowDeletions 206
- AllowEdits 206
- AllowFilters 206
- AllowFormView 199
- AllowLayoutView 199, 263
- AllowMultipleValues 507
- AllowMultiselect 784
- AllowReportView 263
- AllowZeroLength 375
- AllTables 347
- Alt 196
- ALTER DATABASE 639
- ALTER TABLE 962
- ALTER TRIGGER 603
- AlternateBackColor 272
- Analysis Services 585
- And 103
- Anhänge 924
- Anlage 283
- Anlage-Feld 66, 401, 403, 501, 924
- ANSI 547, 749
- ANSI-Standard 92 902
- Ansicht 888, 989
- ANY 922
- API 1107
- ApiViewer 1110
- Append 467, 763
- appendChild 829, 831
- AppendOnly 500
- Application 207, 220, 222, 335
- Application Programmers Interface 1107
- ApplyFilter 70, 212, 273
- AppTitle 980
- ÄQUIVALENZ 103
- Arbeitsmappen 1144
- Arbeitsspeicher 1161
- Arbeitsumgebung 369
- ArcCos 117
- ArcCoTangens 117
- Archiv-Funktion 499
- ArcSin 117
- ArcSinHyperbolicus 117
- Arithmetische Operatoren 101
- Array 91, 92, 1115

- As 74
- As Any 1115
- ASC 110, 463, 546, 943
- ASCII 749
- Assistent 999
- Async 868
- AtEndOfLine 765
- AtEndOfStream 765, 767
- Atn 116
- Attachment 66
- Attribute 811, 814, 837
- Attributes 380, 424, 471, 760, 793, 831, 839
- Auflistungen 324, 357, 360
- Aufrufeliste 138
- AusführenAnwendung 71
- AusführenCode 71
- AusführenDatenmakro 179
- Auslagerungsdatei 1161
- AuslösenFehler 178
- Ausrufezeichen 41
- Ausschneiden 1103
- Auswahlabfrage 382
- AUTO-JOIN 899, 916
- AutoCenter 199
- AutoExec 993
- AutoExec-Makro 173
- Autoincrement 492
- AutoKeys-Makro 171
- AutoSize 237, 252
- AvailableSpace 752
- AVG 944
- Azure 651
- Azure-Server 654

- B**
- BackColor 220, 222, 245, 269, 357
- Backstage-Ansicht 1081
- BackStyle 220, 222, 357
- Backup 612
- BACKUP DATABASE 614
- BaseConnectionString 346
- baseName 831
- Beep 70
- Befehlszeilen-Optionen 997
- BeforeBeginTransaction 213
- BeforeChange 175
- BeforeDelConfirm 212
- BeforeDelete 175
- BeforeInsert 212
- BeforeUpdate 212, 230
- BEGIN CATCH 619
- BEGIN TRY 619
- BeginTrans 408, 448, 471, 974
- BeginTransComplete 472
- BeiFehler 178
- Beispieldatenbank 900
- Benannte Argumente 126
- Benutzer 515
- Benutzerdefinierte Eigenschaften 1030
- Berechnete Spalten 508
- Berechnungsfunktionen 944
- Bereichsgrenzen 93
- Berichtsansicht 261
- Berichtseigenschaften 263
- Berichtseignisse 273
- Berichtsfuß 300
- Berichtskopf 299
- Berichtsmethoden 276
- Berichtsmodule 317
- Betriebssystem 1164
- BETWEEN 907
- Bevel 237
- Bezeichner 75
- Beziehungen 960
- Bild einlesen 1139
- Bild-Ressourcen 255
- Bilder drucken 1140
- Bildschirmauflösung 1161
- Binärdatei 770
- BOF 386, 412, 461, 472
- Bookmark 386, 399, 430, 435, 472
- Bookmarkable 386
- Boolean 73, 83
- BorderColor 220, 223
- BorderStyle 200, 204, 220, 223
- BorderWidth 220, 223
- BottomMargin 220, 225, 293
- BoundColumn 507
- BREAK 591
- BrokenReference 339
- BrowseForFolder 791
- BrowseTo 70, 259
- BuildPath 794
- BuiltIn 332

button 1057
buttonGroup 1063
ButtonName 784
ByRef 124
Byte 73
ByVal 124, 1114

C

CacheSize 386, 472, 541
CacheStart 386, 541
Callback 1052
Callback-Methoden 1101
CameraDeviceType 1142
Cancel 471, 472, 473
CancelBatch 473
CancelEvent 70
CancelUpdate 387, 473
Caption 200, 205, 220, 225
CASE 105, 591
Catalog 487, 524
CDate 86, 122, 152, 295
Change 152, 230, 248
CharToOem 749
ChDir 792
ChDrive 792
checkBox 239, 248, 1064
childNodes 831, 871
Choose 105
Chr 110
Circle 276, 279, 354
Class 1121
Clear 467
Click 210, 230, 239
Clipboard-Objekt 1104
Clone 157, 385, 387, 473
cloneNode 831
Close 70, 208, 387, 449, 471, 765, 792
CloseButton 200, 204
CloseClipboard 1104
CloseCurrentDataBase 342
Cloud 651
Codd 889
Code-Far-Modell 663
Code-Fenster 37, 39
Code-Near-Modell 664
COLLATE 639
Collection 327, 332, 357
ColNameHeader 548
ColorMode 293
Column 242, 765, 793
ColumnCount 242, 507, 976
ColumnHistory 283, 405, 500, 926
Columns.Append 490
ColumnSpacing 293
columnWidthPercent 1088
ColumnWidths 976
COM 1119
COM1 337
comboBox 242, 1066
Command 442, 449, 476
CommandText 449, 451, 472, 476, 495
CommandTimeout 471, 472
CommandType 449, 472, 476
CommitTrans 408, 448, 471
CommitTransComplete 472
CompactDatabase 395, 528, 564
CompactRepair 564
ConnectComplete 447, 472
Connection 442, 445, 471
ConnectionString 445, 471, 474
ConnectionTimeout 448, 471, 541
Const 79
CONSTRAINT 959
Containers 422, 424, 519, 1031
CONTAINS 606, 610
CONTINUE 591
Control 248, 313, 348
ControlBox 200, 204
Controls 218, 248, 325, 337
ControlSource 220, 228, 245, 299
ControlTipText 220, 226
ControlType 220, 228, 248, 313, 325, 347, 348, 357
Convert_TXT_XML 880
ConvertAccessProject 342
Copies 293
Copy 759
CopyFile 794
CopyFolder 794
CopyObject 70
CopyQueryDef 387
CopyTo 860
Cos 116
CoSekans 117
CosinusHyperbolicus 117
CoTangens 117

- CoTaskMemFree 790
 - COUNT 212, 324, 467, 944, 970
 - COUNTER 958
 - CountOfDeclarationLines 331
 - CountOfLines 331
 - CREATE FULLTEXT CATALOG 609
 - CREATE FULLTEXT INDEX 609
 - CREATE INDEX 961
 - CREATE PROCEDURE 598, 962
 - CREATE TABLE 594, 958
 - CREATE VIEW 963
 - CreateAdditionalData 858
 - createAttribute 825, 837, 876
 - createCDATASection 825
 - createComment 825
 - CreateControl 315, 348
 - CreateDatabase 369, 371, 514
 - createDocumentFragment 825
 - createElement 825, 830, 836, 864
 - createEntityReference 825
 - CreateEventProc 331
 - CreateField 374, 380, 500, 503, 507
 - CreateFolder 794
 - CreateForm 312, 348
 - CreateGroup 516, 521
 - CreateGroupLevel 303
 - CreateIndex 377
 - CreateKey 778
 - createNode 825
 - CreateObject 751, 1128, 1173, 1177
 - CreateParameter 472
 - CreateProcess 1159
 - CreateProcessingInstruction 825, 864
 - CreateProperty 406, 503, 507, 981
 - CreateQueryDef 381
 - CreateRelation 380
 - CreateReport 312
 - CreateReportControl 315
 - CreateTableDef 374, 378, 502, 539, 550
 - CreateTextFile 766, 794
 - createTextNode 825
 - CreateUser 516, 521
 - CreateWorkspace 366
 - CStr 110
 - CSV 546, 879
 - CSVDelimited 547
 - CurDir 792
 - Currency 74, 86, 113
 - Current 208, 212, 400
 - CurrentData 347
 - CurrentDb 412, 415
 - CurrentObjectName 339
 - CurrentObjectType 339, 348
 - CurrentProject 345, 475
 - CurrentProject.Path 445
 - CurrentRecord 206
 - CurrentSectionLeft 200
 - CurrentUser 1034
 - CurrentView 200
 - CurrentX 265, 268
 - CurrentY 265, 268
 - Cursor 441, 889
 - CursorLocation 446, 471, 472, 534
 - CursorType 453, 473, 534
 - Cycle 207
- ## D
- DAO 365, 412
 - DAP 66
 - Data Access Pages 66
 - Data Definition Language 882
 - Data Manipulation Language 883
 - Data Query Language 883
 - Database 367, 409, 412, 974
 - Database .NET 706
 - DATABASEPROPERTYEX 687
 - DataChange 212
 - DataEntry 206
 - DataOnly 293
 - DatasheetAlternateBackColor 200
 - DatasheetGridLines 200
 - dataType 831
 - DATE 74, 84, 121, 150, 152, 299, 648, 947
 - DATEADD 122, 948
 - DateCreated 344, 386, 753, 793
 - DATEDIFF 122, 948
 - DateGrouping 271
 - Datei-Öffnen-Dialog 404
 - Dateidialoge 782
 - Dateisystem 752
 - Dateizugriff 750
 - DateLastAccessed 753, 793
 - DateLastModified 753
 - DateModified 344
 - Datenbank 369

- Datenbank analysieren 422
- Datenbank defragmentieren 740
- Datenbank-Nutzer 669
- Datenbankdatei 637
- Datenbankdiagramm 605
- Datenbankkopie 690
- Datenbanknutzer 1034
- Datenbankpasswort 562
- Datenbanksicherung 615
- Datenbankstruktur 641
- Datenbankverwaltung 957
- Datenbankverzeichnis 757
- Datenblattansicht 199
- Dateneigenschaften 264
- Datenfeld 92, 99
- Datenklassenmodul 476
- Datenmakro 174
- Datenmakros 62
- DatensatzBearbeiten 179
- DatensatzErstellen 179, 196
- DatensatzLöschen 179
- Datensatzzeiger 415
- Datensicherheit 622
- Datentypen 73, 83, 96
- Datentypumwandlung 952
- DATEPART 948
- DateSerial 121
- datetime2 648
- datetimeoffset 648
- DateValue 121, 951
- Datum 592, 947, 976, 1161
- Datumsfeld 376
- Datumfunktionen 120
- Datumsteile 971
- Datumsvergleich 909
- Day 121
- dbAppendOnly 383
- dBASE 488, 542
- dbAttachment 502
- dbComplexText 507
- dbConsistent 384
- dbcreator 624
- dbDenyRead 383
- dbDenyWrite 383, 533
- dbEncrypt 370
- DBEngine 365
- dbFixedField 375
- dbInconsistent 383
- DbIclick 210, 230
- dbOptimistic 384
- dbPessimistic 384
- dbReadOnly 384
- dbSeeChanges 384
- dbSQLPassThrough 384
- dbVersion120 370
- DDB 119
- DDE 1130
- DDEExecute 1132
- DDEInitiate 1132
- DDEPoke 1132
- DDERequest 1132
- DDETerminate 1132
- DDL 882, 957
- Deactivate 208, 273
- Debug 360
- Debug.Assert 132
- Debug.Print 131
- Debugger 157
- Debugging 130, 135, 1020
- Declare 1111
- DecodeAttribute 759
- Default 492
- DefaultDatabase 471
- DefaultSize 293
- DefaultValue 220, 375
- DefaultView 200, 261, 263
- definition 831
- DefType 76
- Deklaration 75
- DELETE 212, 387, 393, 433, 458, 467, 473, 552, 759, 936
- deleted 602
- DeleteFile 794
- DeleteFolder 794
- DeleteKey 779
- DeleteLines 331
- DeleteObject 70
- DeleteRule 496
- DeleteValue 780
- Delimited 547
- DESC 463
- Description 492, 511
- Detailbereich 299
- Device 1137
- DeviceInfo 1137
- DeviceManager 1136

DeviceName 293
Diagramm 303
Dictionary 329
Differenz 897
Dim 74, 153
Dir 529
Direktfenster 131
Dirty 206, 212, 214
Disconnect 449, 472
diskadmin 624
DisplayType 1122
DisplayWhen 203, 220, 227
DISTINCT 913, 974
DISTINCTROW. 913
DividingLines 200, 205
Division 102
DLL 1107
DML 882
Do 107
DoCmd 70, 1103
DoCmd.BrowseTo 259
DOCTYPE 814
Document 422
Document Object Model 823
Document Type Definition 810
documentElement 825, 871
Documents 519, 1031
DOM 823
DOMDocument 823, 826, 864, 866
DoMenuItem 70
Doppelte Datensätze 973
Double 73, 83
DQL 882
DrawMode 268, 294
DrawStyle 268
DrawWidth 268
Dritte Normalform 892
Drive 793
DriveExists 794
DriverName 293
Drives 752, 793
DriveType 752
DROP COLUMN 962
DROP TABLE 634, 962
DropDown 242
dropDownElement 1070
DTD 810, 827
Duplex 293

Dynamic 367
Dynamic Link Libraries 1107
Dynamisches Array 127, 794
Dynaset 206, 367, 383, 384

E

Early Binding 311
Echo 70
Edit 387, 392, 412
editBox 1065
EditMode 386, 392, 412, 457, 473
Eigenschaften 34, 198, 319, 322, 406
Eigenschaftenfenster 38
Einbetten 1120
Einfügen 400, 1103
Einführungsbeispiel 50
Eingabemaske 425
Einschränkungen 583
Element 811, 814, 837
Else 105
Embedding 1120
Empty 90
EmptyClipboard 1104
Enabled 220, 226, 412, 435
Endlosformular 245
EndOfRecordset 474
Entwurfsansicht 199, 261
Enum 98
Enumeration 98
EOF 386, 394, 412, 461, 472, 792
Equi-Join 898
Eqv 103
Erase 95
Ereignisse 35, 198
Err.Number 140
Error 90, 139, 142, 214, 273, 470
Errors 443, 621
Erste Normalform 890
EventProcPrefix 220, 227
Events 35
Excel 488, 551, 1144
Excel.Application 1177
Exchange 488
Execute 367, 449, 452, 471, 472, 476, 630, 967
ExecuteComplete 472
ExistKey 777
Exit 107, 230

EXKLUSIV ODER 103
Exp 116
Exponentialfunktion 119
Export 554, 933
ExportXML 856
Expression 510
Extension 760

F

Farbeigenschaften 269
Farbkonstanten 271, 295
FastLaserPrinting 271
Fehlerbehandlung 139, 152
Fehlermeldung 1055
Fehlermodell 616
Fehlersuche 130, 157
Feldtypen 497
FestlegenFeld 178
FestlegenLokaleVar 178
FetchComplete 474
FetchProgress 474
Field 373, 409, 468
Field2 402
FieldChangeComplete 474
Fields 427, 443
File System Objects 750
FileAttr 792
FileCopy 792
FileData 403
FileDateTime 122, 792
FileDialog 404, 783
FileExist 371, 758
FileExists 794
FileExtension 1139
FileFlags 403
FileLen 792
FileName 283, 403, 925
FilePicker 783
Files 759, 793
Fileserver 571
FileSplit 761
FileSystem 752, 793
FileSystemObject 751, 793, 806
FileTimeStamp 403
FileType 403, 925
FileURL 403
FillCache 387
FillColor 269, 270
FillStyle 269, 270, 280
Filter 206, 212, 264, 273, 386, 397, 460, 473
FilterIndex 784
FilterInfos 1143
Filtern 286, 906
FilterName 262
FilterOn 206, 264, 295
FilterOnLoad 206, 264
Find 331, 464, 473
FindClose 799
FindFirst 387, 396, 419
FindFirstFile 799
FindLast 387, 396
FindNext 70, 387, 396, 400
FindNextFile 799
FindPrevious 387, 396
FindRecord 70
Firewall 657, 697
FIRST 944
FirstChild 831, 873
FitToScreen 200
Fix 116, 118
Fixed 113
Folder 753, 754, 806
FolderExists 794
FolderPicker 783
Font 224, 252
FontBold 220, 269
FontItalic 220, 269
FontName 220, 269
FontSize 220
FontUnderline 220
FontWeight 220
For 107, 153
For Each 107, 108, 325
ForeColor 220, 222, 269
ForeignKey 496
ForeignName 380
Form 199, 245, 248, 312, 352
Form_Close 995
Form_Error 143
Format 87, 110, 112, 150, 152, 241, 273, 295, 299, 427
FormatCurrency 87
Formatierte Spalten 938
Forms 324
Formular 199

- Formular-Instanzen 352
 - Formularansicht 199
 - Formularentwurf 36
 - Formulargestaltung 52
 - Formularmodule 317
 - Fortschrittsanzeige 983
 - Forward 367
 - FreeFile 792
 - FreeSpace 752, 793
 - FREETEXT 606, 610
 - Fremdschlüssel 886
 - FSO 750
 - FullPath 332
 - Function 124
 - Funktionen 116, 123
 - FürJedenDatensatz 179
 - FV 120
- G**
- gallery 1073
 - Gebundene Formulare 398
 - General Number 113
 - geography 649
 - geometry 649
 - Geräteigenschaften 1137
 - Gespeicherte Prozedur 888
 - Get 769, 792
 - GetAbsolutePathname 794
 - GetAttr 792
 - GetBaseName 794
 - GetClipboardData 1104
 - GetDeviceCaps 1161
 - GetDrive 794
 - GetDriveName 794
 - Geteilte Ansicht 199
 - GetElementsByTagName 825, 841
 - GetExtensionName 794
 - GetFile 794
 - GetFileName 794, 879
 - GetFolder 753, 754, 794, 806
 - getImage 1059
 - getNamedItem 868, 876
 - GetObject 1128
 - GetOpenFileName 786
 - GetOption 341, 1035
 - GetParentFolderName 794
 - GetPath 879
 - GetPrivateProfileString 772
 - GetProfileString 771
 - GetRegisteredOrganization 1168
 - GetRegisteredOwner 1168
 - GetRows 387, 390
 - GetSaveFileName 786
 - GetSpecialFolder 758, 794
 - GetString 462, 473
 - GetSystemDirectory 1114, 1161
 - GetSystemInfo 1161
 - GetTempName 762, 794
 - GetText 1105
 - GetTickCount 1169
 - GetValue 777
 - GetWindowsDirectory 1161
 - GlobalAlloc 1105
 - Globale Tabelle 643
 - Globale Variable 957
 - GlobalInSameProcess 1120
 - GlobalLock 1105
 - GlobalMemoryStatusEx 1161
 - GlobalOnNewProcess 1119
 - GlobalUnlock 1105
 - GoSub 109
 - GotFocus 208, 230, 273
 - GOTO 109, 153, 592
 - GoToControl 70
 - GoToPage 70, 214
 - GoToRecord 70
 - Grafikbearbeitung 1142
 - Grafikmethoden 276
 - GRANT 630
 - GRANT INSERT 632
 - GridX 200, 264
 - GridY 200
 - Groß-/Kleinschreibung 639, 971
 - GROUP BY 919
 - GroupInterval 303
 - GroupOn 303
 - Groups 525
 - GrpKeepTogether 264
 - Gruppen 515, 624
 - Gruppen ausblenden 1048
 - Gruppenfuß 300
 - Gruppenkopf 299
 - Gruppieren 178, 299
 - Gruppierung 283
 - GUID 332, 497

Gültigkeitsbereiche 80

H

Haltepunkt 160
hasChildNodes 831
HashModule 207
Hauptindex 377, 886
Hauptregisterkarten ausblenden 1046
HAVING 919
Height 220
HelpContextID 207, 220, 271
HelpFile 207, 271
hierarchyid 649
Hilfe 42
HinzufügenMenü 71
Historie 500
History-Feld 926
HistoryCount 927
HKEY_CURRENT_USER 773
HKEY_LOCAL_MACHINE 773
HOUR 121, 947
Hourglass 70
HTML 224, 489, 807
HTML-Fragment 719
HWnd 207
hWndAccessApp 992
Hyperlinkfeld 376

I

IDE 36
IDENTITY 642, 958
idMso 1044
If 105
IgnoreNulls 378
Iif 105, 390
ImageFile 1139, 1142
imageMso 1044
ImageProcess 1142
Imp 103
IMPLIKATION 103
ImportXML 856
IN 910
Increment 492
Index 377, 386, 395, 435, 886
Indexdefinition 493
Indexes 493

Indizes 373, 376
InfoMessage 472
INI-Datei 771
Initial Catalog 446
InitialFileName 784
Initialize 332, 354
InitialView 784
INNER JOIN 303, 898, 915
Input 763, 792, 794
InputBox 147, 149, 153
InSameProcess 1120
InSelection 220, 227
INSERT INTO 639, 930
insertBefore 831
inserted 602
InsertLines 331
InsertText 331
InsideHeight 200
InsideWidth 200
Installation SQL Server 573
Instanzenbildung 322
InStr 110, 149, 943, 974
InStrRev 111
Int 116, 118
Integer 73, 83
Integerfeld 376
Integrated Security 446
Integration Services 584
IntelliSense 60
InvalidateControl 1053
IP-Filter 662
IP-Freigabe 666
IPmt 120
IRR 119
Is 428
Is Nothing 310
IS NULL 911
ISABOUT 611
IsArray 91
IsBroken 332
IsClipboardFormatAvailable 1104
IsDate 122
IsInsert 179
IsLoaded 344
IsMissing 801
IsNull 390, 412
IsolationLevel 448, 471
IsReady 793

IsRootFolder 793
IsSystemNT 1167
Item 328
ItemAdded 333
ItemLayout 293
ItemRemoved 333
ItemsAcross 293
ItemSizeHeight 293
IXMLDOMAttribute 876
IXMLDOMNode 871, 876

J

Jahr 2000 86
Jet OLEDB:Allow Zero Length 492
Jet OLEDB:AutoGenerate 492
Jet OLEDB:AutoGenerate 498
Jet OLEDB:Column Validation Rule 492
Jet OLEDB:Column Validation Text 492
Jet OLEDB:Compressed UNICODE Strings 492
Jet OLEDB>Create Link 552
Jet OLEDB:Hyperlink 492
Jet OLEDB:Link Datasource 552
Jet OLEDB:Link Provider String 552
Jet OLEDB:Remote Table Name 552
JET_SCHEMA_USERROSTER 561
Jet-Engine 570
JetEngine 529
Jetzt() 196
JOIN 111, 794, 898, 914
JRO 529

K

KeepTogether 303
Kennwort 562
Key 237, 496
KeyCode 434
KeyDown 210, 230, 275, 434
KeyPress 210, 230, 239, 275
KeyPreview 207
KeyUp 210, 230, 241, 275
Kill 529, 792, 878, 1139
Kind 333
Klasse 354
Klassenbibliothek 334
Klassenmodul 241, 320, 354
Kombinationsfeld 242, 419

Kommentare 41, 811
Konfiguration 1038
Konstanten 73
Konstantendeklaration 79
Kontrollstrukturen 105
Kopfzeile 980
Kopieren 1103
Kreuztabellenabfrage 953
Kubikwurzel 102
Kuchendiagramm 1177
Künstlicher Schlüssel 887

L

labelControl 1057
LAST 944
LastChild 831, 873
LastModified 386, 430
LastUpdated 386
Late Binding 311
Laufwerke 752
Layout 1084
Layoutansicht 199, 261
layoutContainer 1087, 1093
LayoutForPrint 200, 264
LBound 95
LCASE 111, 943
Leerzeichen 636, 843
LEFT 111, 220, 943
LEFT-JOIN 898, 915
LeftMargin 220, 225, 293
LEN 111, 149, 943
Library 1019
LIKE 908
LIMIT 742
Line 276, 277, 764, 765, 793
Line Input 792
Lines 331
LineSpacing 225
LinkChildFields 217, 245, 248
Linking 1120
LinkMasterFields 217, 245, 248
Listenfeld 419, 476
Lizenznehmer 1167
Load 208, 273, 825, 826
LoadCustomUI 1039
LoadCustomUI als Alternative 1042
LoadFromFile 402, 861, 866

LoadPicture 252, 1059
LoadXML 825, 866
Loc 792
LocalDB 567, 570, 579
Lock Read 750
Lock Write 750
Locked 220, 226, 239, 248
LockEdits 386, 531
LockNavigationPane 985
LockType 455, 473, 534
LOF 764, 769, 792
Log 116
Logarithmus 119
Login-Account 669
LoginTimeout 541
Lokal-Fenster 132
Lokale Tabelle 643
Long 73, 83
LongPtr 1108
Loop 107
Löschen 400
LöschenMakroFehler 178
LostFocus 208, 230, 273
LPT1 294
Istrcat 790
LTRIM 111, 943

M

m:n-Beziehung 404, 895
Major 333
Makro 35, 163
Makroaktionen 173
Makrokonvertierung 61
Managed Add-In 1011
MarshalOptions 473
master 583
MAX 944
Maximize 70
MaxRecords 473
MaxScanRows 548
Me 207, 313
mehrwertige Felder 928
Meldung 71
Memofeld 283, 376, 499
menu 1075
Menü 1037
Menü-Assistent 1001
Menüband ändern 1055
MenuBar 207
Metadaten 743
Metasprache 807
Methoden 34, 198, 319, 322
Methods 34
Microsoft Access 15.0 Object Library 1051
Microsoft Access 16.0 Object Library 335
Microsoft ActiveX Data Objects 6.1 Library 438
Microsoft Excel 1177
Microsoft Office 16.0 Object Library 335
Microsoft Word 1173
Microsoft XML 824
Microsoft.ACE.OLEDB.16.0 439, 445
MID 111, 943
MIN 944
Minimize 70, 985
MinMaxButtons 200, 204
Minor 333
MINUTE 121, 947
MIRR 120
MkDir 756, 792
Mod 102, 152
Modal 200
Mode 448, 471
Model 583
Module 207, 330
Modules 330
Modulo 102
MONTH 121, 947
MouseDown 210, 230, 275
MouseMove 210, 230
MousePointer 233
MouseUp 210, 230
MouseWheel 210, 275
Move 214, 230, 387, 462, 473, 759
MoveComplete 466, 474, 476
MoveFile 794
MoveFirst 387, 412, 417, 474
MoveFolder 794
MoveLast 387, 412, 416, 474
MoveNext 387, 394, 417, 457, 462, 474
MovePrevious 387, 394, 412, 417, 474
MoveSize 70
msado15.dll 438
MSDataShape 441
msdb 583
MSDN-Abo 658

MsgBox 144, 153, 430
msoFileDialogFilePicker 783
msoFileDialogFolderPicker 783
msoFileDialogOpen 783
MSysDb 1032
Multi Value Field 66, 404
Multiuserzugriff 530
Multivalue-Feld 503
Mustereigenschaften 269
MVF 66, 404, 503

N

Nach Aktualisierung 175
Nach Einfügung 175
Nach Löschung 175
Nachfolgeereignisse 180
Nachschlage-Assistent 405
NachschlagenDatensatz 178
Name 207, 220, 333, 344, 386
NavigateTo 988
NavigationButtons 200, 205
Navigationsbereich 984
Navigationskategorien 986
Navigationsschaltflächen 479
Navigationssteuerelement 257
NEGATION 103
Netzwerkzugriff 577
New 318, 352, 1128
NewPassword 372
Next 107
NextRecordset 474
NextSibling 831, 873
NoData 273
nodeFromID 825
nodeName 831
Nodes 806
nodeTxFedValue 831
nodeType 831
nodeTypeString 831
NoMatch 386, 396, 400, 419, 435
Normalisieren 889
Northwind 583
Not 103
Nothing 352, 751
NOW 122, 947
NPer 120
NPV 120

NULL 90, 911
NULL-Value 888, 945
Nullable 492
Nutzerpass 563

O

Object 74, 311, 1128
Object Linking and Embedding 1120
ObjectFrame 1121
Objects 34
Objekt-Assistent 1003
Objekt-Katalog 1127
Objekte 34
Objekteigenschaften 53
Objektfeld 1121, 1124, 1177
Objektkatalog 334
Objekttypen 309
Objektvariablen 309
ODBC 535
ODBC-Datenquellen-Administrator 535
ODBC-Verbindung 537
ODBCTimeout 541
ODER 103
OEM 547
OemToChar 749
Office 365 658
OldValue 220, 228
OLE 1119
OLE DB 441
OLE-Automation 1127, 1156
OLE-Client 1120
OLE-Server 1120
OLEDB 439
OLETypeAllowed 1122
On Error GoTo 140
On Error Resume Next 140
OnConnect 213
OnDataAvailable 868
OnDisconnect 213
OnNewProcess 1119
OnReadyStateChange 868
Open 208, 445, 471, 474, 763, 765, 792, 794
Open-Dialog 783
OpenArgs 262, 263
OpenAsTextStream 766
OpenClipboard 1104
OpenCurrentDataBase 342

OpenDatabase 371, 533, 538
OPENFILENAME 786
OpenForm 70, 72, 992
OpenModule 70
OpenQuery 70
OpenRecordset 383, 387, 412, 419, 427, 966
OpenReport 70, 262, 295
OpenSchema 471, 561
OpenTable 70
OpenTextFile 766, 794
Operatoren 100, 103
Optimierung 556
Optimistic Locking 530
Option Base 92
Option Compare Database 57
Option Explicit 57, 77
Optional 126
Optionale Argumente 125
OptionGroup 248, 295
Options 450
Optionsfeld 248, 479
Or 103
ORDER BY 596, 918
OrderBy 206, 264
OrderByOn 206, 264
OrderOnLoad 206, 264
Orientation 293
OUTER-JOIN 898
Output 764, 794
OutputTo 71, 288
ownerDocument 831

P

Pack 394
Page 212, 273
PageCount 473
PageFooter 263
PageHeader 263
Pages 248
PageSize 473
Paint 273
Painting 207
PaletteSource 200
Panel 252
PanelClick 236
PanelDbClick 236
PaperBin 293
PaperSize 293
Paradox 488
ParamArray 126
Parameter 469, 476
Parameter-Arrays 126
Parameterabfrage 381, 495
Parameters 381, 444, 450
Parent 220, 222, 314, 344
ParentCatalog 498
ParentFolder 793
ParentNode 831, 873
ParentRecordset 402
parsed 831
parseError 825, 827
Parser 813
Passwort 514
Paste 1126
Path 346, 757, 793
PDF-Datei 287
PERCENT 113, 912
PercentPosition 386
Performance 697
Permissions 519, 522
Pessimistic Locking 530
Pfad 760
PI 79, 117, 813
Picture 200, 205
PictureAlignment 201, 264
PicturePages 264
PictureSizeMode 201, 264
PictureTiling 201, 264
PictureType 201, 264
PIVOT 954
Platzhalter 908
Platzhalterzeichen 636
Pmt 120
PopUp 201
Potenzieren 102
PPmt 120
pragma table_info 743
Prepared 472
Preserve 95, 794
preserveWhiteSpace 825
PreviousSibling 831, 873
Primärschlüssel 886
Primary 378
Primary Index 886
Primary Key 959

- PrimaryKey 494
 - Print 273, 276, 280, 765, 792
 - Printer 288
 - Printers 289
 - PrintOut 71
 - PrintQuality 293
 - Private 80, 128, 1119
 - ProcBodyLine 331
 - ProcCountLines 331
 - processadmin 624
 - Processing Instructions 810, 813
 - ProcOfLine 331
 - ProcStartLine 331
 - Progressbar 983
 - ProjectType 346
 - Projekt-Explorer 39
 - Prolog 810
 - Properties 34, 344, 360, 406, 443, 448, 492, 552, 981, 1031, 1138
 - Property 329, 467
 - Property Get 320, 354
 - Property Let 320, 354
 - ProtokollierenEreignis 179
 - Provider 471, 535
 - Prozedur 495
 - Prozeduren 123
 - Prozessoranzahl 1161
 - PSet 276, 278
 - Public 80, 128
 - PublicNotCreatable 1119
 - Put 769, 792
 - PV 119
- Q**
- QBColor 270
 - Quadratwurzel 102
 - Quadrieren 119
 - Query Analyzer 581
 - QueryDef 381
 - QueryDefs 422
 - QueryPerformanceCounter 1169
 - QueryTimeout 541
 - Quit 71, 343, 992, 996
- R**
- RAISEERROR 617
 - Random 768
 - Randomize 116, 118
 - Rate 120
 - RDS 441
 - Read 765
 - ReadAll 765
 - ReadLine 765, 767
 - ReadText 861
 - readyState 825
 - Recalc 214
 - Rechteverwaltung 964
 - Record 1116
 - Record-Locking 530
 - RecordChangeComplete 474
 - RecordCount 386, 388, 460, 473
 - RecordLocks 206, 264
 - RecordsAffected 450, 453, 930
 - RecordSelectors 201, 205
 - Recordset 367, 383, 398, 412, 443, 474, 889
 - Recordset2 402
 - RecordsetChangeComplete 474
 - RecordsetClone 206, 399
 - RecordsetType 206
 - RecordSource 206, 245, 264, 398
 - ReDim 94, 153, 794
 - ReDim Preserve 129
 - Reference 332, 1026
 - References.AddFromFile 1022
 - References.Remove 1023
 - Refresh 215, 467
 - RefreshLink 552
 - RefreshRecord 71
 - RefreshTitleBar 980
 - REG_BINARY 774
 - REG_DWORD 774
 - REG_SZ 774
 - RegCloseKey 776, 1168
 - RegCreateKeyEx 776
 - RegDeleteKey 776
 - RegDeleteValue 776
 - RegFlushKey 776
 - Register 248
 - RegisterDatabase 537
 - Registerkarte 1049
 - Registerkarten ausblenden 1047
 - Registerkartenansicht 1034
 - Registrierdatenbank 773
 - RegOpenKey 1168
 - RegOpenKeyEx 776

RegQueryValueEx 776, 1168
RegSetValueEx 776
RelatedTable 496
Relation 380
Relationen 379, 496
Remotezugriff 577
Remove 328
removeChild 831
RemoveItem 242
RemoveReferenz 333
Rename 71
Repaint 215
RepaintObject 71
Replace 111
replaceChild 831
ReplaceLine 331
Replikation 66
Replizieren 555
Report 295, 312, 354
Reportansichten 261
Reporting Services 584
ReportName 262
Reports 324
Requery 71, 215, 387, 474
Required 375, 378
Reset 792
Resize 208, 273
Resources 256
Restartable 386
Restore 71, 985
RESTORE DATABASE 614
Resume 140
Resync 474
Retreat 273
REVOKE 630
RGB 270
Ribbon 1037
RibbonCreator 1100
RibbonName 1040
Rich-Text-Feld 282, 502
RIGHT 111, 943
RIGHT-JOIN 898, 915
RightMargin 221, 225, 294
Rmdir 792
Rnd 116, 117
Rollback 408, 974
RollbackTrans 448, 471
RollbackTransComplete 472

Rollen 624
RootFolder 793
Round 116, 150
RowSource 507
RowSourceType 242, 507
RowSpacing 294
RTF-Datei 288
RTRIM 111, 943
RunCommand 1104
Rundung 118
RunMacro 71
RunSQL 71
Runtime 1024
Runtime-Modus 996

S

Save 71, 461, 474, 878
SaveAs-Dialog 783
SaveFile 1139
SaveToFile 402, 861
Scale 276
ScaleHeight 265
ScaleLeft 265
ScaleMode 265
ScaleTop 265
ScaleWidth 265
Scanner 1134
Scanner-Assistent 1141
ScannerDeviceType 1142
Schaltflächen einfügen 1049
Schema.ini 547
Schieberegler 415
Schlüssel 886
Schnellzugriffsleiste 1056
Schrifeigenschaften 269
Schweregrade 616
Scientific 113
Screen 231
ScrollBars 201, 205
SECOND 121, 947
secondColumn 1087
Section 201, 203, 221, 227, 245, 272
securityadmin 624
Seed 492
Seek 387, 435, 769, 792
Seitenansicht 262
Seitenfuß 300

- Seitenkopf 299
- Sekans 117
- SELECT 105, 555, 904
- SELECT INTO 933
- SelectedItems 784
- SelectNodes 841
- SelectObject 71, 984
- SelectSingleNode 834, 841
- Selektion 897
- SelLength 149, 239, 1104
- SelStart 149, 239, 1104
- SelText 239
- SendenEMail 179
- SendKeys 211, 434
- SendMessage 1115
- SendObject 71
- separator 1061
- SerialNumber 752
- Set 309, 352
- SET ROWCOUNT 635
- SetAttr 792
- SetClipboardData 1104
- SetDisplayedCategories 989
- SetFilter 71
- SetFocus 149, 214, 231, 239, 1104
- SetMenuItem 71
- setNameItem 876
- SetOption 341, 1035
- SetOrderBy 71
- SetParameter 71
- SetPermissions 525
- setupadmin 624
- SetValue 779
- SetWarnings 71
- SetzenWert 71
- Sgn 116
- Share Deny Read 560
- Share Deny Write 560
- Shared 750
- SharedResources 255
- SHBrowseForFolder 790
- Shell 1159
- ShellExecute 1159
- Shellfunktionen 801
- SHFileOperation 801
- SHFILEOPSTRUCT 801
- SHGetPathFromIDLList 791
- ShortcutMenu 207
- ShortcutMenuBar 207, 221, 226
- ShortPath 754, 793
- ShowAcquireImage 1139
- ShowAcquisitionWizard 1141
- ShowAllRecords 71
- ShowDatePicker 221
- ShowDeviceProperties 1137
- ShowPhotoPrintingWizard 1140
- ShowToolBar 71
- ShowWindow 991
- Sicherheitscenter 46
- Sicherheitseinstellungen 43
- Sicherheitsmodell 623
- Sicht 495, 595
- Sichten 962
- SimpleText 252
- Sin 116
- Single 73, 83
- SinusHyperbolicus 117
- Size 793
- SizeMode 1122
- SizeToFit 231
- Skip 766
- SkipLine 766, 861
- Slider 415
- SLN 120
- Snapshot 367, 383, 385
- Snippet 719
- Sort 386, 395, 463, 473
- Sortieren 463
- SortOrder 303
- SOUNDEX 716
- Source 473
- SourceDoc 1123, 1126
- SourceItem 1122
- SourceObject 217
- SourceTableName 539
- sp_addlogin 630
- sp_addrole 631
- sp_detach_db 640
- sp_droplogin 630
- sp_password 630
- sp_rename 642
- SPACE 111, 943
- Spalten auswählen 904
- Spaltenverlauf 926
- Special 1125
- SpecialEffect 203, 221, 223

- Split 111, 794
 - Split View 199
 - splitButton 1078
 - SQL 881
 - SQL Azure 652
 - SQL Distributed Management Objects 585
 - SQL Server 2014 568
 - SQL Server 2014 Express 566
 - SQL Server Browser-Dienst 633
 - SQL Server Express 577
 - SQL Server Konfigurationsmanager 582
 - SQL Server Management Studio 579
 - SQL Server Profiler 581
 - SQL Server-Agent 584
 - SQL Server-Authentifizierung 623
 - SQL Server-Datentypen 649
 - SQL-Abfrageprogramm 974
 - SQL-Pass-Through 382
 - SQL-Passthrough 966
 - SQL-Server 446
 - sqlcmd 580, 699
 - SQLEXPRESS 575
 - SQLite 702
 - SQLite Administrator 708
 - sqlite_master 743
 - sqlite_sequenz 744
 - SQLite-Datentypen 712
 - Sqlite3.exe 737
 - SQLOLEDB 446
 - SQLServer 586
 - Sqr 116
 - Stammelement 812
 - Standard 113
 - Standarddialogfelder 143
 - startFromScratch 1047
 - StartUpShowStatusBar 982
 - State 471, 472, 473
 - Static 82, 128, 149
 - Status 473
 - StatusBar 235, 252
 - StatusBarText 221, 226
 - Statusleiste 252
 - Statuszeile 982
 - STDEV 944
 - STDEVP 944
 - Step 107
 - Steuerelemente 218, 956
 - Steuerelemente-Assistent 1007
 - Stifteigenschaften 268
 - StopAlleMakros 71
 - StopMakro 71
 - StoppAlleMakros 179
 - StoppMakro 178
 - Stored Procedure 597, 888
 - Str 111
 - Stream 860, 866
 - STRING 74, 87, 111, 943
 - Stringaddition 88
 - Stringverarbeitung 110
 - StrReverse 111
 - Style 237, 252
 - Sub 124
 - SubFolders 754, 793
 - SubForm 245, 248
 - SUBSTRING 636
 - Suchen 464
 - Suchleiste 989
 - SUM 303, 944
 - Supports 456, 474
 - SW_HIDE 992
 - Switch 105
 - SYD 119
 - Symbolleiste 1037
 - Synchronisieren 696
 - sysadmin 624
 - SysCmd 339, 1024
 - System-Verzeichnis 757
 - SystemFolder 758
 - Systemressourcen 1161
 - Systemtabellen 989
- ## T
- T-SQL 587
 - Datentypen 648
 - Variablen 589
 - TabControl 248
 - TabDelimited 547
 - Tabellen 373
 - Tabellenblatt 1152
 - Tabellendefinition 489
 - Tabellenspalte 642
 - TabIndex 221, 226
 - Table 367, 383, 384, 490
 - TableDef 373, 406, 409
 - TableDefs 422, 539

Tables 519
TabStop 221, 226
Tag 207, 221, 225
Tan 116
taskFormGroup 1086
Tastaturbefehle 71
Tastaturstatus 252
TCP/IP-Protokoll 577
Teilstring 636
Temp-Verzeichnis 757
tempdb 583
Tempfile 762
Temporäre Tabelle 595
TemporaryFolder 758
TempVars 295, 340, 957
Terminate 332, 354
text 831
TextAlign 221, 224
TextBox 239
Textdatei 546, 762, 794
Textdaten 879
Textfeld 149, 239, 376
TextFormat 221, 224, 239, 282, 503
TextHeight 276, 281
TextWidth 281
Then 105
TIME 121, 648, 947
Timer 121, 150, 214, 248, 273, 435
TimerInterval 150, 207, 248, 430
TimeSerial 121
Timestamp 744
TimeValue 121
Titelleiste. 980
To 93
toggleButton 1062
TOP 221, 635, 912
TopMargin 221, 294
TopMargin, 225
Tortendiagramm 280
TotalSize 752, 793
Transact-SQL 587
Transactions 386
Transaktionen 408
TransferDatabase 71, 554
TransferSpreadsheet 71, 1145
TransferText 71
TRANSFORM 954
TreeView 422, 424, 806, 870

Trigger 601, 888
Trim 111, 974
TSQL 636
TXT 546
Type 96, 331, 344, 386, 793, 1117
TypeName 89
TypeOf 428
Typisierte Datei 768
Typkennzeichen 75

U

Überwachungs-Fenster 134
Überwachungsausdruck 161
UBound 95, 804
UCase 111, 943, 974
Uhrzeit 592, 1161
Uhrzeitvergleichen 910
UND 103
Undo 215, 217, 231
UndoBatchEdit 213
Ungebundene Recordsets 460
Unicode 749
UNION 917
Unique 377
uniqueidentifier 648
Unload 208, 273
UnspecifiedDeviceType 1142
Unterabfrage 920, 925
Unterbrechungsmodus 132
Unterformular 217, 245
Unternavigation 1089
Unterverzeichnis 754
Until 107
Updatable 386, 391, 412, 428
UPDATE 387, 392, 412, 430, 457, 474, 935
UpdateBatch 459, 474
User 516
UserDefined 1031
Users 525
USysApplicationLog 177
USysRegInfo 1000
USysRibbons 1041
UTF-16 814, 864
UTF-8 814, 864

V

Val 111, 354
validateOnParse 825, 827
ValidateOnSet 375
ValidationRule 375, 386
ValidationText 375, 386
Value 221, 228, 239, 248, 389, 412
VALUES 930
VAR 944
varbinary 649
Variablen 73
Variablendeklaration 74
Variant 74, 89
VARP 944
VarType 89, 148, 427
VBA 56
VBA7 1109
vbCrLf 88
vbCurrency 427
vbEmpty 148, 427
vbNull 148
Verb 1122
Verbindungszeichenfolge 439
Vereinigung 896
Vergleichsoperatoren 104
Verknüpfen 1120
Verlaufsverfolgung 405
Verschlüsseln 527
Version 471
Verweise 1026
Verzeichnis 753, 757
Verzeichnis erzeugen 756
Verzeichnisbaum 806
Verzeichnisdialog 790
Verzweigungen 105
VideoDeviceType 1142
View 595, 888, 962
Views 495
ViewsAllowed 201
Visible 221, 226
Visual Basic For Applications 335
Volltextabfrage 606
Volltextabfragen 717
VolumeName 752, 793
Vor Änderung 175
Vor Löschung 175
Vorabereignisse 179

W

Währungswert 86
WaitForSingleObject 1159
Webdatenbank 652
WEEKDAY 121, 152, 947
WEIGHT 611
Wenn 178
WhatsThisButton 201
WHERE 906, 914
WhereCondition 262
WHILE 107, 591
WIA 1134
wiaEventDeviceConnected 1136
Width 201, 221
WillChangeField 474
WillChangeRecord 465, 474
WillChangeRecordset 474
WillConnect 472
WillExecute 472
WillMove 474
Win.ini 771
Win64 1108
WindowHeight 201
WindowMode 262
Windows Azure 651
Windows Image Acquisition Automation Layer 1134
Windows Live 658
Windows-Authentifizierung 623
WindowsFolder 758
WindowWidth 201
With 97
WITH OWNERACCESS OPTION 513
WithEvents 323, 447, 476, 868, 1136
Wochenende 152
Word.Application 1173
Workspace 366, 974
Write 766, 792, 861
WriteBlankLines 766
WriteLine 766, 767
WritePrivateProfileString 772
WriteProfileString 771
WriteText 861
Wurzel 119

X

XML 807, 831, 866
XML-Datei 864

XML-Dokumente 829
XML-Editor 1099
XML-Export 745
XML-Fehlerprüfung 827
XML-Integration 846
XML-Transformation 853
XMLDOMAttribute 823
XMLDOMNamedNodeMap 823
XMLDOMNode 823, 831
XMLDOMNodeList 823
XMLDOMParseError 823
Xor 103
XPath 840
XSD-Schema 817
XSLT 853

Y

YEAR 121, 947

Z

Zählerfeld 375

Zeichenfolgen 908
Zeichenkette 88
Zeichenketten 588
Zeichenkettenfunktionen 110
Zeichensatz 814
Zeilenumbruch 40, 88
Zeit 947, 976
Zeitangabe 150
Zeitfunktionen 120
Zeitmarke 376
Zeitmessungen 1169
Zufallswerte 497
Zufallszahlen 117
Zugriffsrechte 372
Zugriffsschutz 512
Zusammengesetzter Schlüssel 887
Zustandsvariablen 481
Zustandswechsel 481
Zweite Normalform 890
Zwischenablage 1103