

Datenbankprogrammierung in Visual-Basic 5.0

Christian Zahler

7 Zugriff auf Datenbanken

Visual Basic ist **zweifellos** eine datenbankorientierte Programmiersprache.

Der Zugriff ist auf drei Arten von Datenbanken möglich:

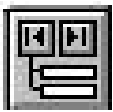
- **Microsoft Jet Databases:** Dies sind Datenbanken mit der Erweiterung *.MDB, wie sie zum Beispiel von Access hergestellt werden
- **ODBC-basierte Datenbanken:** siehe später!
- **Installierbare ISAM-Datenbanken** (ISAM = index sequential access method = indexsequentielle Zugriffsmethode)

Einer der schnellsten Wege, auf externe Daten zuzugreifen, besteht in der Methode, eine externe Tabelle an eine bereits existierende *.MDB-Datei „anzuhängen“. Ein derartiges Beispiel wird nachstehend im Kapitel ODBC erklärt.

Es können folgende Datenbanken angesprochen werden:

- Btrieve-Datenbanken
- dBASE-Datenbanken
- Lotus Tabellen
- Microsoft Excel Tabellen/Arbeitsmappen
- Microsoft Foxpro-Datenbanken
- Microsoft Jet Datenbanken (Access usw.)
- Paradox Datenbanken
- Text-Dokumente

Das wichtigste Steuerelement für Daten ist **Data**.



Für Zugriff auf Daten in Datenbanken

Nach dem Aufbringen dieses Steuerelements auf die Form wählt man die Datenbank und die Tabelle aus:

```
Data1.DatabaseName = „C:\KUNDEN.MDB“
Data1.RecordSource = „firmen“
```

Dann verbindet man die Steuerelemente, die für die Anzeige der Daten gedacht sind (Textfelder, Labels, PictureBox, Image) mit dem Daten-Steuerelement, etwa:

```
Text1.DataSource = Data1
```

Anschließend werden Steuerelementen bestimmte Felder zugewiesen:

```
Text1.DataField = "name"
```

8 ODBC (Open DataBase Connectivity)

8.1 WOSA (Windows Open Services Architecture):

Microsoft entschloss sich, für Dienste, die zwar denselben Zweck haben, aber unterschiedliche Datenstrukturen verwenden, eine Standardschnittstelle zur Verfügung zu stellen. Die Philosophie dahinter: Der Programmierer soll von der Anpassung seines Produkts an viele spezielle Hardware-Eigenschaften entbunden werden.

Derzeit stehen innerhalb dieses Konzepts folgende Schnittstellen zur Verfügung:

- **MAPI (Messaging Application Programming Interface):** Schnittstelle für elektronische Post- und Nachrichten-Dienste. Ein Programmierer, der auf Mail-Produkte zugreifen will, muss daher nur mehr auf diese Schnittstelle zugreifen, die dann die Systemaufrufe auf die spezielle Syntax des Mail-Programms konvertiert. So kann etwa sowohl MS-Mail als auch cc:Mail usw. unterstützt werden.
- **ODBC (Open Database Connectivity):** regelt den Zugriff auf Datenbanken.

In Zukunft ist geplant, dieses Konzept durch weitere Schnittstellen (z.B. für Sicherheit etc.) zu ergänzen.

WOSA verwendet „Dynamic Link Libraries“ (*.DLL-Dateien), um unterschiedliche Anwendungsprogramme während des Betriebs miteinander und mit den Systemdiensten zu verbinden.

8.2 ODBC-Datenbankzugriff

Mit einem ODBC-Treiber können Visual Basic-Programme auf **jede Datenbank** zugreifen. Dafür bestehen folgende Möglichkeiten:

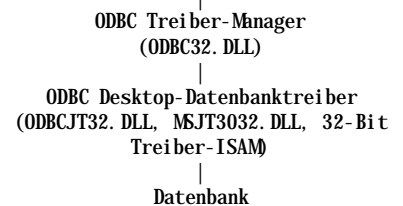
- Man verwendet eine ODBC-Tabelle oder ein View-Objekt der „Microsoft Jet Database“ und verwendet die Methode **OpenRecordset**, um die Tabelle anzusprechen.
- Direktes Öffnen der ODBC-Datenbank mit der Methode **OpenDatabase**
- Erzeugen eines **Querydef**-Objektes, das auf eine ODBC-Datenbank zeigt
- Verwenden der „Data“-Steuerung, um entweder eine ODBC-Tabelle zu öffnen oder die Datenbank direkt anzusprechen
- Verwendung der ODBC-API (enthalten im „ODBC programmer’s toolkit“)

Es müssen die ODBC-Treiber installiert sein und ein „Data Source Name“ (DSN) muss vergeben worden sein.

Mit Hilfe der Microsoft ODBC Desktop-Datenbanktreiber können Sie über die ODBC-Schnittstelle (Open Database Connectivity) Datenbanken öffnen und abfragen. Diese Treiber wurden für die Verwendung unter Microsoft Windows 95 oder höher bzw. Windows NT 3.51 oder höher entwickelt. Nur 32-Bit-Anwendungen werden unter Windows 95 oder später unterstützt, während unter Windows NT 3.51 oder später 16-Bit- und 32-Bit-Anwendungen unterstützt werden. Die ODBC Desktop-Datenbanktreiber umfassen 32-Bit-Treiber für Microsoft Access, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox und Text, und keine 16-Bit-Treiber.

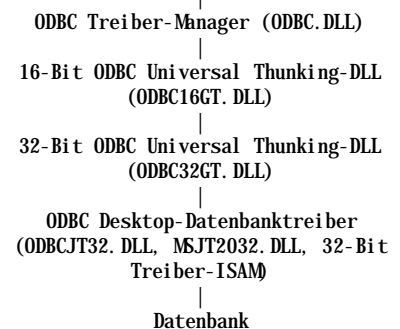
Die Anwendungsprogramm/Treiber-Architektur sieht für Anwendungen unter **Windows 95** (oder höher) wie folgt aus:

Anwendung (32-Bit)

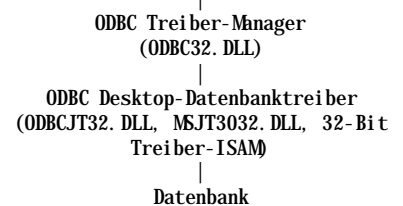


Die Anwendungsprogramm/Treiber-Architektur sieht für Anwendungen unter Windows NT ab Version 3.51 wie folgt aus:

Anwendung (16-Bit)



Anwendung (32-Bit)



Mit dem Microsoft Jet-Datenbankmodul können Sie auf folgende externen Datenquellen zugreifen.

- Microsoft Jet-Datenbanken, darunter Datenbanken, die mit früheren Versionen von Microsoft Jet erstellt wurden.
- Microsoft FoxPro-Datenbanken, Version 2.0, 2.5, 2.6, 3.0 und DBC (Datenbank-Container).
- dBASE-Datenbanken, Version III®, dBASE IV® und dBASE 5.0.
- Paradox-Datenbanken, Version 3.x, 4.x und 5.x.
- Microsoft Excel-Arbeitsblätter, Version 3.0, 4.0, 5.0, 7.0 und 8.0.
- Lotus-Tabellen, Version WKS, WK1, WK3 und WK4.
- ASCII-Textdateien im Tabellenformat.
- Tabellendaten in HTML-Dateien (Hyper-Text Markup Language).
- Ordner in Microsoft Exchange-Mailboxen.

Während der Installation der ISAM-Treiber erstellt das Setup-Programm in der Windows-Registrierung Schlüssel für die folgenden Datenquellen:

- Microsoft FoxPro 2.0, 2.5, 2.6, 3.0 und DBC
- dBASE III, dBASE IV und dBASE 5.0
- Paradox, Version 3.x, 4.x und 5.x
- Microsoft Excel 3.0, 4.0, 5.0 und 8.0
- Lotus WK1, WK3 und WK4
- ASCII-Textdateien im Tabellenformat
- HTML-Dateien
- Microsoft Exchange 4.0

Die ODBC Desktop-Datenbanktreiber beinhalten folgende ISAM- und Setupdateien:

Treiber	32-Bit-ISAM	Versionen
Microsoft Access	Keine (integriert in MSJT3032.DLL)	Microsoft Access, Version 7.0
Microsoft Access	MSRD2X32.DLL	Microsoft Access, Versionen 1.0, 1.1, 2.0
dBase	MSXB3032.DLL	dBase, Versionen III, IV und V (dBASE für Windows)
Microsoft Excel	MSXL3032.DLL	Microsoft Excel, Versionen 3.0, 4.0, 5.0 (Zu Versionen vor Version 3.0 siehe Anmerkung)
Microsoft FoxPro	MSXB3032.DLL	Microsoft FoxPro, Versionen 2.0, 2.5, 2.6
Paradox	MSPX3032.DLL	Paradox, Versionen 3.5, 4.0, 4.5, 5.x (Paradox für Windows)
Text	MSTX3032.DLL	ASCII-Text

Anmerkung: Der Microsoft Excel-Treiber kann keine Microsoft Excel-Dateien lesen, die in einem früheren Format als dem der Version 3.0 vorliegen. Solche Dateien können mit einer späteren Version von Microsoft Excel geöffnet und im Format der Version 3.0, 4.0 oder 5.0 gespeichert werden.

Beispiel 6: Datenaustausch mit Excel (B06)

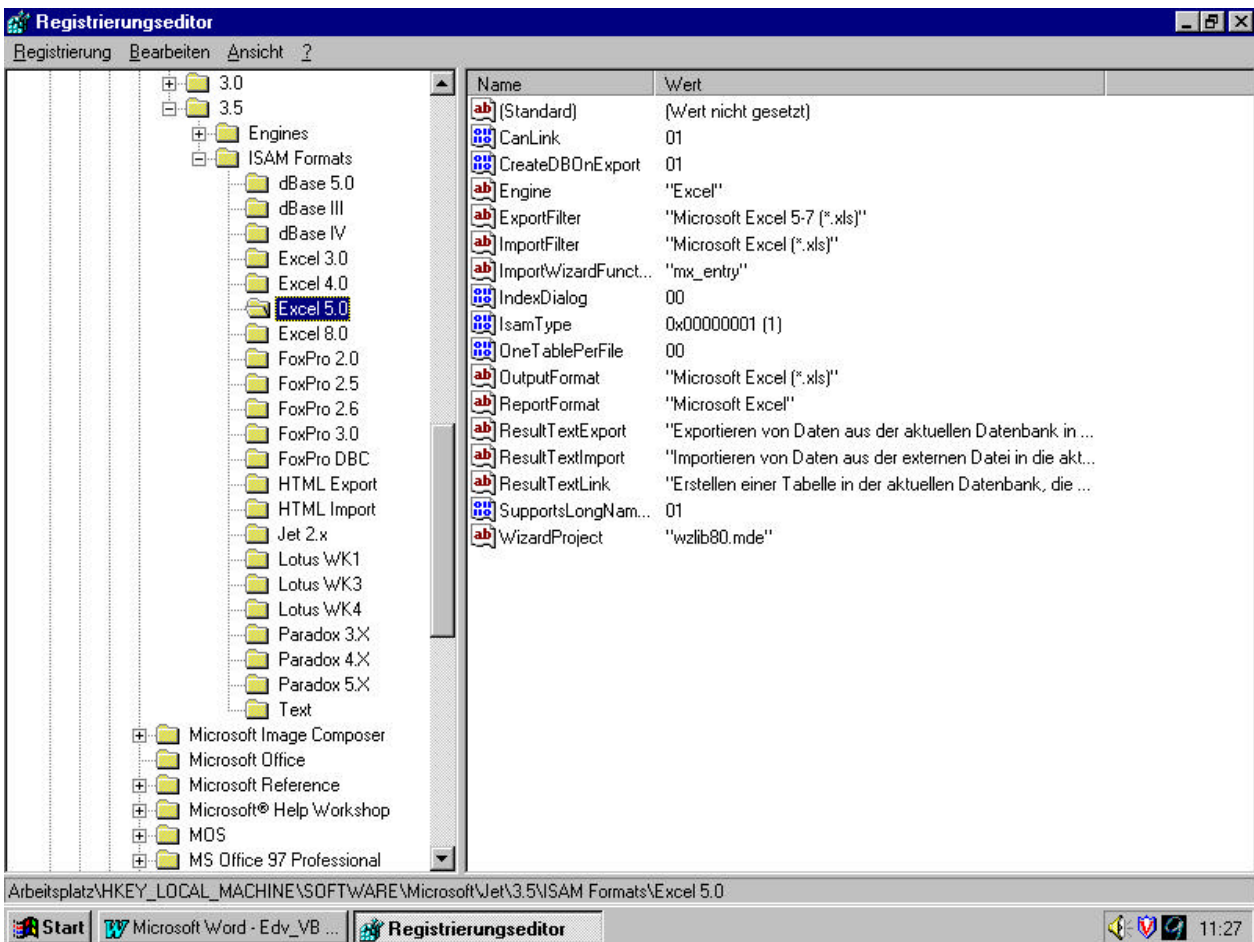
Es soll auf eine extern vorhandene EXCEL-Tabelle zugegriffen werden.

Wichtig: Es muss der ISAM (Index Sequential Access Method = indexsequentieller Zugriff) für Excel in der Registry vorhanden sein. ISAM beschreibt die Organisation von Daten in einem File. Bei dieser Organisation ist sowohl der direkte als auch der sequentielle Zugriff auf Daten zugelassen

Dazu überprüft man folgenden Eintrag in der Registry von Windows 95 (**Bild**):

```
HKEY_LOCAL_MACHINE
Software
Microsoft
Jet
3.5
ISAM Formats
```

Die Tabelle wurde unter dem Dateina-



men TEST.XLS gespeichert und sieht folgendermaßen aus:

	A	B
1	Name	Anzahl
2	Floppy 3,5"	3
3	HDD Seagate 4 GB	2
4	HDD Conner 1,2 GB	7
5	Miro Magic-Grafikkarte	5
6	S3-Grafikkarte	2
7	Matrox Millennium 4 Ml	12
8		

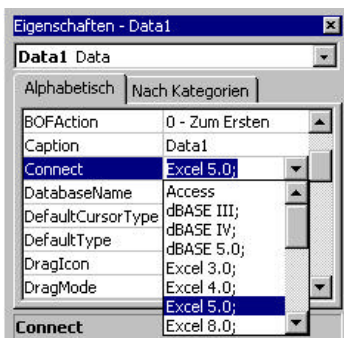
Der Zugriff ist nun über das Data-Steuer-element möglich. Dazu erstellt man sich eine Form wie folgt:



1. Möglichkeit: Automatisiert (B06A)

Beim Data-Steuer-element müssen folgende Eigenschaften geändert werden:

Connect	gibt die Art der Verbindung an (Standard ist Access)	Excel 5.0
DataBaseName	gibt den Pfad für die Datenquelle an	C:\...\TEST.XLS
RecordSetType	Datenbankstruktur= Tabelle	0 - Table
RecordSource	Bezeichnung der Excel-Tabelle (wichtig: \$-Zeichen!)	Tabelle1\$



Bei den beiden Textfeldern müssen folgende Eigenschaften geändert werden:

DataField	gibt die Feldbezeichnung an (in der 1. Zeile der EXCEL-Tabelle enthalten)	Name bzw. Anzahl
DataSource	Auf welches Daten-Steuer-element soll sich die Anzeige beziehen	Data1

Hinzufügen neuer Datensätze:

Die Eigenschaft EOFAction des Daten-Steuer-elementes muss auf „2“ gesetzt werden.

Mögliche Einstellungen für die EOFAction-Eigenschaft:

- EOFAction Operation
- 0 Setzt den aktuellen Datensatz auf die letzte Position und verhindert, daß sich der Benutzer hinter das Ende der Datensatzgruppe bewegen kann
- 1 Setzt den aktuellen Datensatz auf EOF und deaktiviert die Schaltfläche für „Nächster Datensatz“ im Daten-Steuer-element
- 2 Überprüft den letzten Datensatz und ruft automatisch die **AddNew**-Methode auf. Das Daten-Steuer-element wird anschließend sofort auf den neuen Datensatz gesetzt.

2. Möglichkeit: Codierung (B06B)



Vorgangsweise: Es wird nicht direkt auf die Excel-Tabelle zugegriffen, sondern über eine Access-Datenbank, die eine Tabelle enthält, die mit der ursprünglichen Excel-Tabelle in Zusammenhang steht.

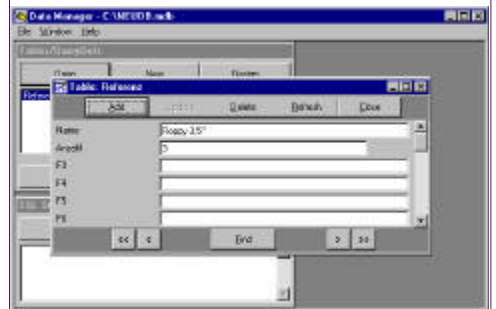
Zuerst muss mit Access eine leere Datenbank (im vorliegenden Beispiel mit dem Pfad C:\NEUDB.MDB) angelegt werden.

```
Private Sub Form_Load
    Dim Tabelle As TableDef
    Dim Datenbank As Database

    Set Datenbank = _
        OpenDatabase("C:\NEUDB.MDB")
    Set Tabelle = _
        Datenbank.CreateTableDef("Referenz")
    Tabelle.Connect = _
        "Excel 7.0; DATABASE=C:\TEST"
    Tabelle.SourceTableName = "Tabelle1$"
    ' wenn nur Bereich gewünscht:
    ' ... = "Tabelle1$A1:E15"
```

```
Datenbank.TableDefs.Append Tabelle
ConnectSource = True
Text1.DataField = "Name"
Text2.DataField = "Anzahl"
End Sub
```

Die Tabelle kann mit dem **Data-Manager** (erreichbar über das Menü **Add-Ins - Data-Manager**) angesehen und bearbeitet werden:



Bemerkung: Beim **zweiten** Aufruf entsteht eine Fehlermeldung, da die Tabelle bereits besteht. Verbessern Sie das Programm derart, dass kein Fehler mehr entstehen kann!

Visual Basic-Kurs in den PCNEWS-Ausgaben

Den kompletten Kurs können Sie in Form von PDF-Dokumenten aus dem PCNEWS-Web beziehen:

http://pcnews.at/ins/pcn/<xx>/_pdf/n<xx><yyyy>.pdf

xx ist die Nummer der PCNEWS-Ausgabe, **yyyy** die erste Seite des Beitrags mit führender Null, jeweils aus der folgenden Tabelle:

Kap	xx	yyyy	Seite	Einführung in Visual Basic 5.0
1	58	082	82	Einleitung
2	58	082	83	Grundlagen der Programmierung
3	59	036	38	Grafikprogrammierung in Visual Basic
4	60	063	63	Das Standarddialogobjekt
5	60	064	64	Dateizugriff
6	62	049	49	Datenaustausch: OLE und DDE
7	65	0072	72	Zugriff auf Datenbanken
8	65	0072	72	ODBC (Open DataBase Connectivity)
9	63	107	107	Einbinden von EXE-Dateien in Visual Basic-Programme
10	63	107	107	Verwendung von Dynamic Link Libraries (DLLs)
11	64	0078	78	Erstellung von Online-Hilfe-Dateien
A	59	036	36	Befehlsübersicht
B	60	066	66	ANSI-Zeichensatz