

# ACCESS

## IM UNTERNEHMEN

## TASTENKOMBINATIONEN IM GRIFF

Kennen Sie alle Access-Tastenkombinationen?  
Wir zeigen alle bekannten und unbekanntenen  
Helfer! (ab S. 2)



### In diesem Heft:

#### **RIBBON FÜR DAS RICH-TEXT-FELD**

Zeigen Sie ein Ribbon mit allen notwendigen Befehlen für das Rich-Text-Feld an.

#### **UNTERBERICHTE NACH WAHL**

Blenden Sie Unterberichte in einem Hauptbericht nach Wunsch ein und aus.

#### **ACCESS UND SQL SERVER- SICHERHEIT, TEIL 3**

Alles was Sie über die SQL Server-Authentifizierung wissen müssen.

SEITE 26

SEITE 30

SEITE 39

## Schneller zum Ziel

**Irgendwann im Laufe meiner Schulzeit habe ich einmal einen Schreibmaschinenkurs gemacht. Eine sinnvolle Investition! Meine Kenntnisse blieben zwar lange ungenutzt, bis irgendwann das Schreiben umfangreicherer Texte auf dem Programm stand und ich mit etwas Übung schnell mit hoher Geschwindigkeit tippen konnte. Wenn da nicht nur immer der Griff zur Maus wäre, um Befehle auszuführen ... Doch auch hier naht Abhilfe: Wir zeigen Ihnen, wie Sie fast alle Aufgaben unter Access schnell mit der Tastatur erledigen können.**



Bei der Recherche zum Beitrag **Tastenkombinationen in Access** (ab Seite 2) gab es immer wieder Aha-Momente: Mir sind im Laufe der Zeit doch einige Tastenkombinationen verborgen geblieben. Wenn Sie also die Hände an der Tastatur lassen wollen, schauen Sie sich an, was alles mit der Tastatur möglich ist!

Verknüpfungen zwischen Tabellen sind schnell durch das Einrichten von Nachschlagefeldern oder über das Beziehungen-Fenster eingerichtet. Aber was, wenn Sie einmal wissen wollen, über welche Felder die Beziehung zwischen zwei Tabellen realisiert wurde, ohne das Beziehungen-Fenster nutzen zu können? Wenn diese Aufgabe beispielsweise per VBA erledigt werden muss? Dann erfahren Sie alles Notwendige im Beitrag **Beziehungen zwischen Tabellen/Abfragen ermitteln** ab Seite 10. Hier schauen wir uns sogar noch an, wie Sie herausfinden, welche Beziehungen es zwischen den Tabellen gibt, die in Abfragen zusammengefasst wurden!

Für den ergonomischen Umgang mit **Ja/Nein**-Feldern haben uns eine Lösung ausgedacht, bei der Sie mit zwei Listenfeldern auf der einen Seite die Datensätze mit dem **Ja**-Wert und auf der anderen die Datensätze mit dem **Nein**-Wert im **Ja/Nein**-Feld anzeigen können. Wie Sie das programmieren, zeigen wir Ihnen im Beitrag **Ja/Nein-Daten selektieren mit zwei Listenfeldern** (ab S. 18).

Das Rich-Text-Feld liefert zwar ein Popup-Menü mit, das erscheint, sobald der Benutzer den zu formatierenden Text markiert hat – aber dieses ist nicht sofort sichtbar und manchmal wird es wieder ausgeblendet, bevor man es

zum Formatieren verwenden konnte. Also beschreiben wir im Beitrag **Ribbon für Rich-Text-Felder** (ab Seite 26), wie Sie die benötigten Befehle gut sichtbar im Ribbon anzeigen, sobald der Benutzer das Rich-Text-Feld aktiviert.

Wer umfangreiche Datenmengen in Berichten anzeigt, verwendet dabei vermutlich auch Unterberichte. Im Beitrag **Unterberichte ein- und ausblenden** ab Seite 30 zeigen wir, wie Sie vor dem Öffnen des Berichts die anzuzeigenden Unterberichte auswählen.

Bernd Jungbluth setzt seine Beitragsreihe zum Thema SQL Server-Sicherheit fort. Unter dem Titel **SQL Server-Security – Teil 3: SQL Server-Authentifizierung** finden Sie ab Seite 39 alles Wissenswerte rund um die Verwendung der SQL Server-Authentifizierung.

Oft gefragt werden wir danach, wie man PDF-Dokumente im Webbrowser-Steuerelement von Access anzeigen kann. Wie das gelingt, beschreiben wir ab Seite 63 im Beitrag **PDF-Dokumente im WebBrowser-Steuerelement**.

Und schließlich liefern wir mit dem Beitrag **.accde-Datenbank per VBA erstellen** auf Seite 74 noch eine Funktion zum Erstellen von **.accde**-Datenbanken per VBA-Code.

Viel Spaß beim Lesen!

A handwritten signature in black ink, appearing to read 'A. Minhorst'.

Ihr André Minhorst

# Tastenkombinationen in Access

Die Arbeit mit einer Anwendung wird umso effizienter, je schneller Sie bestimmte Schritte durchführen können. In Zeiten von Tastatur und Maus gibt es einige Aufgaben, die sich nur mit der Maus erledigen lassen, aber mit der Tastatur gehen die meisten Erledigungen schneller von der Hand – mit Ausnahme zum Beispiel des gezielten Anklickens von Steuerelementen, die nicht auf die Schnelle per Tastenkombination erreichbar sind. Für die anderen Fälle stellen wir Ihnen in diesem Beitrag alle gängigen (und auch einige weniger bekannte) Tastenkombinationen für die Arbeit mit der Benutzeroberfläche von Access vor.

## Begriffsklärung

Wir reden in diesem Beitrag synonym von Feld und Steuerelement, weil sich die meisten Tastenkombinationen sowohl auf Felder in der Datenblattansicht (von Tabellen, Abfragen und Formularen) als auch in der Formularansicht von Formularen einsetzen lassen.

## Standard-Tastenkombinationen

Die folgenden Tastenkombinationen kennen die meisten von Ihnen wahrscheinlich – allein, weil sie auch in vielen anderen Anwendungen funktionieren:

- **Tabulator:** Verschiebt den Fokus auf das nächste Steuerelement.
- **Umschalt + Tabulator:** Verschiebt den Fokus auf das vorherige Feld.
- **Eingabetaste:** Verschiebt den Fokus auf das nächste Feld.
- **Strg + Eingabetaste:** Fügt in Textfeldern einen Zeilenumbruch ein. Wenn die Eigenschaft **Eingabetastenverhalten** den Wert **Neue Zeile** aufweist, genügt die Eingabetaste.
- **Strg + C:** Kopiert den markierten Text oder das markierte Element in die Zwischenablage.
- **Strg + X:** Schneidet den markierten Text oder das markierte Element aus und fügt es in die Zwischenablage ein.
- **Strg + V:** Fügt den Inhalt der Zwischenablage ein, wenn der Inhalt zum Ziel passt.
- **Strg + F:** Öffnet den **Suchen**-Dialog.
- **Strg + Z:** Macht die vorherige Aktion rückgängig. Dies ist auf 20 Aktionen begrenzt.
- **Strg + A:** Selektiert den kompletten Text des aktuell markierten Elements oder alle Elemente des aktuellen Objekts.
- **Umschalt + F2:** Öffnet das Zoom-Fenster und zeigt den Inhalt des aktuell markierten Steuerelements oder der Eigenschaft darin an (siehe Bild 1).

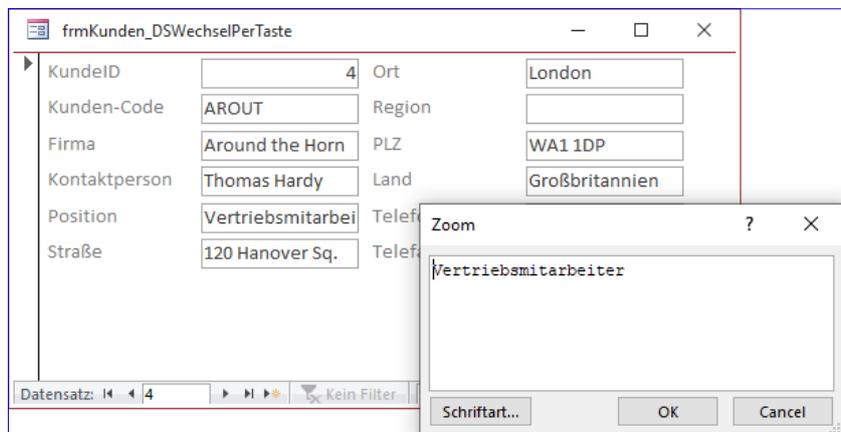
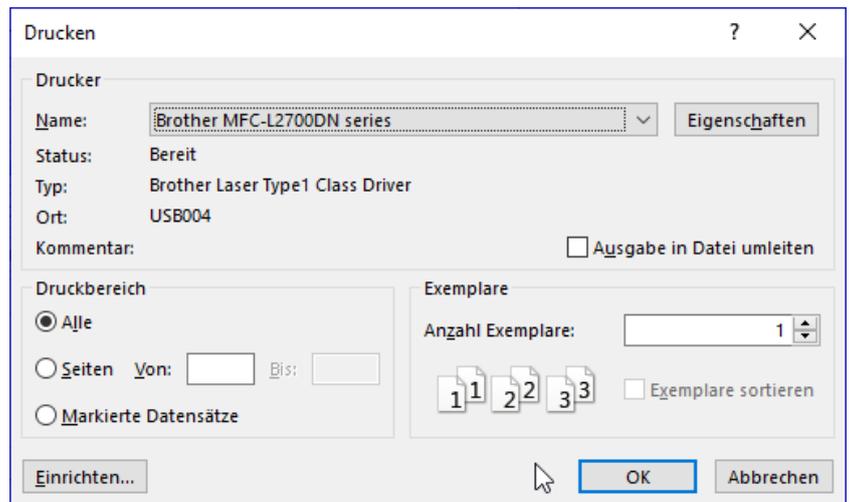


Bild 1: Das Zoom-Fenster von Access

- **Strg + P**: Öffnet den **Drucken**-Dialog (siehe Bild 2).
- **Leertaste**: Aktiviert oder deaktiviert Kontrollkästchen.
- **F7**: Startet die Rechtschreibprüfung.
- **Escape**: Verwirft aktuelle ungespeicherte Änderungen am zuletzt bearbeiteten Feld (einfache Betätigung) oder dem zuletzt bearbeiteten Datensatz (zweifache Betätigung).



**Bild 2:** Der **Drucken**-Dialog

- **Alt + F4**: Schließt die Access-Anwendung.
- **F1**: Zeigt das Hilfe-Fenster an.

### **Datum, Zeit, vorherigen Feldinhalt oder Standardwert einfügen**

Die folgenden Tastenkürzel können Sie in gebundenen Feldern und zumeist auch in ungebundenen Feldern nutzen:

- **Strg + ;**; beziehungsweise **Strg + Umschalt + ;**: Fügt das aktuelle Datum in das aktive Feld ein.
- **Strg + :**; beziehungsweise **Strg + Umschalt + .**: Fügt die aktuelle Uhrzeit in das aktive Feld ein.
- **Strg + '**; beziehungsweise **Strg + Umschalt + #**: Trägt den Wert des Feldes aus dem vorherigen Datensatz ein. Sehr praktisch, wenn Sie öfter Werte aus vorherigen Datensätzen übernehmen wollen, und ist viel schneller als Copy and Paste (nur in gebundenen Feldern)
- **Strg + Alt + Leertaste**: Fügt den Standardwert für das aktuelle Feld ein. Hilfreich, wenn Sie bereits begonnen haben, den Standardwert durch einen anderen Wert zu ersetzen und dann feststellen, dass Sie doch den Standardwert eingeben möchten.

### **Texte markieren**

Wenn Sie sich mit der Einfügemarke in einem Textfeld befinden, möchten Sie diese vielleicht schnell zum Anfang oder zum Ende des Textfeldes bewegen, ausgehend von der aktuellen Position um ganze Wörter nach links oder rechts navigieren oder auch dabei direkt die überfahrenen Texte markieren. All das gelingt mit den folgenden Tastenkombinationen in Feldern mit bearbeitbaren Textinhalten:

- **Nach links**- oder **Nach rechts**-Taste: Bewegt die Einfügemarke um ein Zeichen nach links oder nach rechts.
- **Umschalt + Nach links** oder **Umschalt + Nach rechts**: Bewegt die Einfügemarke um ein Zeichen nach links oder nach rechts und markiert die überfahrenen Zeichen direkt.
- **Strg + Nach links** oder **Strg + Nach rechts**: Bewegt die Einfügemarke um ein Wort nach links oder rechts beziehungsweise, wenn sich die Einfügemarke gerade mitten in einem Wort befindet, zum Wortanfang oder Wortende.
- **Umschalt + Strg + Nach links** oder **Umschalt + Strg + Nach rechts**: Bewegt die Einfügemarke um ein Wort nach links oder rechts beziehungsweise, wenn sich die Einfügemarke gerade mitten in einem Wort befindet,

zum Wortanfang oder Wortende und markiert dabei die überfahrenen Worte.

- **Nach oben-** oder **Nach unten-**Taste: Bewegt die Einfügemarke in die darüber oder darunter befindliche Zeile.
- **Umschalt + Nach oben-** oder **Umschalt + Nach unten-**Taste: Bewegt die Einfügemarke in die darüber oder darunter befindliche Zeile und markiert den Text zwischen der vorherigen und der aktuellen Position.
- **Bild nach oben:** Bewegt die Einfügemarke an die gleiche Position in der obersten Zeile.
- **Bild nach unten:** Bewegt die Einfügemarke an die gleiche Position in der untersten Zeile.
- **Strg + Bild nach oben:** Bewegt die Einfügemarke an die gleiche Position in der obersten Zeile und markiert den Text zwischen der vorherigen und der aktuellen Position.
- **Strg + Bild nach unten:** Bewegt die Einfügemarke an die gleiche Position in der untersten Zeile und markiert den Text zwischen der vorherigen und der aktuellen Position.
- **Pos1:** Bewegt die Einfügemarke an den Anfang der Zeile.
- **Ende:** Bewegt die Einfügemarke an das Ende der Zeile.
- **Umschalt + Pos1:** Bewegt die Einfügemarke an den Anfang der Zeile und markiert die Zeichen zwischen der vorherigen und der aktuellen Position.
- **Umschalt + Ende:** Bewegt die Einfügemarke an das Ende der Zeile und markiert die Zeichen zwischen der vorherigen und der aktuellen Position.
- **Strg + Umschalt + Pos1:** Bewegt die Einfügemarke an den Anfang des Textes (auch mehrzeilig) und markiert

die Zeichen zwischen der vorherigen und der aktuellen Position.

- **Strg + Umschalt + Ende:** Bewegt die Einfügemarke an das Ende des Textes (auch mehrzeilig) und markiert die Zeichen zwischen der vorherigen und der aktuellen Position.
- **F2:** Wenn man in ein Feld gelangt und dieses wird aufgrund der Einstellungen komplett markiert, wechseln Sie mit **F2** zur Anzeige der Einfügemarke und umgekehrt.
- **F4:** Diese Taste klappt ein Kombinationsfeld auf.
- **Strg + A:** Markiert den vollständigen Text des Steuerungselements.
- **Zurück/Backspace:** Löscht die Markierung oder ein Zeichen links von der Einfügemarke.
- **Entf:** Löscht die Markierung oder ein Zeichen rechts von der Einfügemarke.

### Mit Datensätzen in der Formularansicht arbeiten

Die folgenden Tastenkombinationen unterstützen Sie beim Navigieren in den Datensätzen in Formularen in der Formularansicht:

- **Strg + Bild nach oben:** Verschiebt den Datensatzzeiger in der Formularansicht auf den vorherigen Datensatz. Halten Sie die Tastenkombination gedrückt, werden die Datensätze schnell durchlaufen.
- **Strg + Bild nach unten:** Verschiebt den Datensatzzeiger in der Formularansicht auf den vorherigen Datensatz. Halten Sie die Tastenkombination gedrückt, werden die Datensätze schnell durchlaufen.
- **Eingabetaste/Tabulator** vom letzten Feld der Aktivierreihenfolge: Springt zum ersten Feld des nächsten

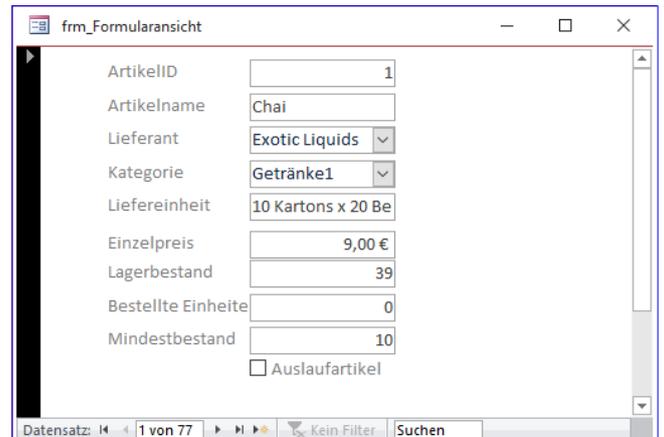
Datensatzes. Gilt nur, wenn die Eigenschaft **Zyklus** den Wert **Alle Datensätze** enthält.

- **Umschalt + Tabulator** vom ersten Feld der Aktivierreihenfolge: Springt zum letzten Feld des vorherigen Datensatzes. Gilt nur, wenn die Eigenschaft **Zyklus** den Wert **Alle Datensätze** enthält.
- **Strg + A**: Markiert den vollständigen Datensatz durch Markieren des Datensatzmarkierers (siehe Bild 3).

### Mit Datensätzen in der Datenblattansicht arbeiten

Für die Datenblattansicht gelten einige spezielle Tastenkombinationen. So können Sie mit **Strg + Bild nach oben/Bild nach unten** zum Beispiel nicht den Datensatz wechseln.

- **Nach rechts/Nach links**: Verschiebt den Fokus zum nächsten Feld auf der rechten oder linken Seite.
- **Nach unten/Nach oben**: Verschiebt den Fokus auf das darüber/darunter liegende Feld und somit zum nächsten/vorherigen Datensatz.
- **Strg + Bild nach oben**: Setzt die Markierung auf das erste Feld auf der rechten Seite, das in der aktuellen Ansicht nicht sichtbar ist und verschiebt somit die Spalten nach links. Ist das letzte Feld bereits sichtbar, wird der Fokus auf dieses Feld verschoben.
- **Strg + Bild nach unten**: Setzt die Markierung auf das erste Feld auf der linken Seite, das in der aktuellen Ansicht nicht sichtbar ist und verschiebt somit die Spalten nach rechts. Ist das erste Feld bereits sichtbar, wird der Fokus auf dieses Feld verschoben.
- **Alt + F5**: Aktiviert das Feld zur Eingabe des anzuzeigenden Datensatzes (siehe Bild 4).



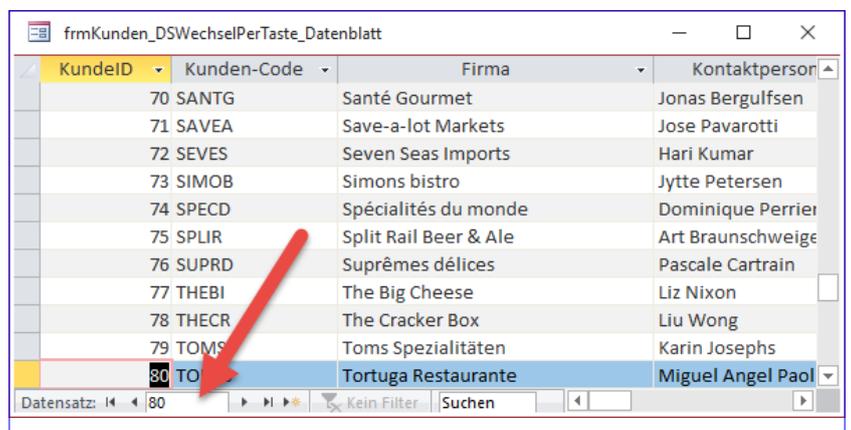
**Bild 3:** Aktivierter Datensatzmarkierer nach **Strg + A**

- **Strg + A**: Markiert alle Datensätze des Datenblatts.

### Mit Datensätzen in verschiedenen Ansichten arbeiten

Die folgenden Tastenkombinationen funktionieren sowohl in der Formularansicht als auch in der Datenblattansicht:

- **Nach rechts/Nach links** ausgehend vom markierten Feld: Bewegt den Fokus zum nächsten/vorhergehenden Feld. Befindet sich der Fokus auf dem ersten oder letzten Feld der Aktivierreihenfolge, wird er auf das erste Feld des nächsten beziehungsweise das letzte Feld des vorherigen Datensatzes bewegt.



**Bild 4:** Aktivieren des Feldes zur direkten Eingabe einer Datensatznummer

## Beziehungen zwischen Tabellen/Abfragen ermitteln

Im Beitrag »Berichte und Unterberichte konfigurieren« haben wir die Aufgabe, für Kombinationen aus Tabellen und Abfragen herauszufinden, ob eine Beziehung zwischen den beiden Datenquellen besteht und über welche Felder diese realisiert wird. In diesem Fall nutzen wir dies, um automatisch die Verknüpfungsfelder für die Datensatzquellen von Haupt- und Unterberichten zu ermitteln. Wir erhalten also die Tabelle/Abfrage des Hauptberichts und des Unterberichts und benötigen nun die Felder, über welche die Beziehung zwischen den beiden Datenquellen hergestellt wird. Wie das gelingt, zeigt der vorliegende Beitrag.

Als Beispiel für diesen Beitrag verwenden wir die Tabellen der Süd Sturm-Datenbank, einen Ausschnitt des Datenmodells sehen Sie in Bild 1.

Das Parentfeld heißt **KundeID**, das Childfeld ebenfalls. Genau diese Informationen wollen wir mit der geplanten Funktion ermitteln.

### Aufgabenstellung

Wir benötigen eine Funktion, die für zwei gegebene Tabellen oder Abfragen die folgenden Daten herausfindet:

### Definition der benötigten Funktion

Die Funktion soll also die folgenden Eingangsparameter enthalten:

- Name der Parenttabelle
- Name des Parentfeldes
- Name der Childtabelle
- Name des Childfeldes

Nochmal die Definition von Parent- und Childtabellen: Die Parenttabelle ist die Tabelle, deren Primärschlüsselfeld an der Beziehung beteiligt ist, die Childtabelle ist die Tabelle, deren Fremdschlüsselfeld an der Beziehung beteiligt ist.

Bei der Beziehung zwischen den Tabellen **tblKunden** und **tblBestellungen** ist also **tblKunden** die Parenttabelle und **tblBestellungen** die Childtabelle.

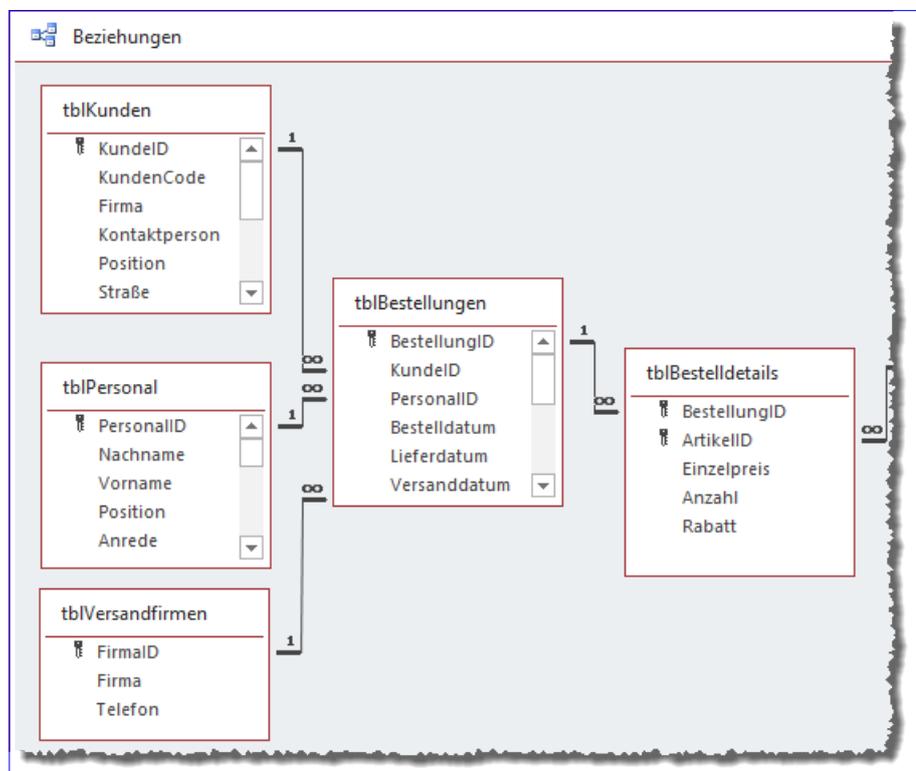


Bild 1: Datenmodell mit Beispielveknüpfungen

- Erste Tabelle (**strTableOrQuery1**)
- Zweite Tabelle (**strTableOrQuery2**)

Außerdem benötigen wir diese Ausgangsparameter für die Funktion:

- Parenttabelle (**strParenttable**)
- Parentfeld (**strParentfield**)
- Childtabelle (**strChildtable**)
- Childfeld (**strChildfield**)

Schließlich soll die Funktion noch einen Funktionswert zurückliefern, der angibt, ob überhaupt eine Beziehung gefunden wurde – dazu verwenden wir den Datentyp **Boolean**. Der Kopf der Funktion sieht also schon einmal so aus:

```
Public Function GetReference(strTableOrQuery1 As String,
strTableOrQuery2 As String, strParenttable As String,
strParentfield As String, strChildtable As String,
strChildfield As String) As Boolean
```

Den Rest der Funktion sehen Sie in Listing 1. Die Funktion stellt zunächst alle Werte der Rückgabeparameter auf eine leere Zeichenkette ein (dies geschieht nur, weil wir meh-

```
Public Function GetReference(strTableOrQuery1 As String, strTableOrQuery2 As String, strParenttable As String,
strParentfield As String, strChildtable As String, strChildfield As String) As Boolean
Dim db As DAO.Database
Dim rel As DAO.Relation
Set db = CurrentDb
strParenttable = ""
strParentfield = ""
strChildtable = ""
strChildfield = ""
For Each rel In db.Relations
If rel.Table = strTableOrQuery1 And rel.ForeignTable = strTableOrQuery2 Then
strParenttable = rel.Table
strChildtable = rel.ForeignTable
strParentfield = rel.Fields(0).Name
strChildfield = rel.Fields(0).ForeignName
GetReference = True
Exit Function
End If
If rel.Table = strTableOrQuery2 And rel.ForeignTable = strTableOrQuery1 Then
strParenttable = rel.Table
strChildtable = rel.ForeignTable
strParentfield = rel.Fields(0).Name
strChildfield = rel.Fields(0).ForeignName
GetReference = True
Exit Function
End If
Next rel
End Function
```

**Listing 1:** Funktion zum Ermitteln von Beziehungsdaten zweier Tabellen

rere Testaufrufe hintereinander durchführen und die Werte der Variablen in den Testaufrufen sonst beim erneuern Aufruf beibehalten würden).

Dann durchläuft die Funktion alle **Relation**-Objekte der aktuellen Datenbank, die wir mit der **Relations**-Auflistung des **Database**-Objekts aus **CurrentDb** erhalten. In dieser **For Each**-Schleife prüfen wir zunächst, ob der Name der mit **strTableOrQuery1** übergebenen Tabelle mit der Eigenschaft **Table** des **Relation**-Objekts übereinstimmt und der Name aus **strTableOrQuery2** mit dem Wert der Eigenschaft **ForeignTable**. **Table** enthält immer den Namen der Parenttabelle, **ForeignTable** den Namen der Childtabelle.

Ist das der Fall, füllt die Funktion einfach die vier Rückgabeparameter mit folgenden Eigenschaften des **Relation**-Objekts:

- **strParenttable**: erhält den Wert der Eigenschaft **Table**
- **strChildtable**: erhält den Wert der Eigenschaft **ForeignTable**
- **strParentfield**: erhält den Wert der Eigenschaft **Fields(0).Name**
- **strChildfield**: erhält den Wert der Eigenschaft **Fields(0).ForeignName**

Warum **Fields(0)**? Es kann sein, dass zwei Tabellen über mehr als eine Beziehung verknüpft sind. Dieser Fall sollte aber in einem sauber formulierten Datenmodell nicht vorkommen, daher berücksichtigen wir nur eine Beziehung.

Wenn die Parameter gefüllt sind, stellen wir noch den Funktionswert auf **True** ein und verlassen die Funktion mit **Exit Function**.

Da es auch sein kann, dass die Parenttabelle mit dem Parameter **strTableOrQuery2** und die Childtabelle mit dem Parameter **strTableOrQuery1** übergeben wurde,

prüfen wir in einer zweiten **If...Then**-Bedingung auch noch den umgekehrten Fall. Auch hier tragen wir dann die entsprechenden Werte in die Rückgabeparameter ein und verlassen dann die Funktion.

### Testen der Funktion

Um die Funktion zu testen, haben wir eine Prozedur mit einigen Testaufrufen programmiert. Diese finden Sie in gekürzter Fassung in Listing 2.

Diese Prozedur ruft die Funktion auf und übergibt beliebige Kombinationen von Tabellen. Liefert die Funktion in diesen Aufrufen den Wert **True** zurück, prüfen wir in einer **If...Then**-Bedingung, ob die vier Rückgabeparameter die erwarteten Werte zurückliefern.

Falls nicht, geben wir eine entsprechende Meldung im Direktbereich des VBA-Editors aus. Wenn schon das Funktionsergebnis entgegen den Erwartungen **False** lautet, geben wir ebenfalls eine Meldung im Direktbereich aus.

Am Ende haben wir auch noch einen Test für die Kombination zweier Tabellen eingefügt, für die es keine Beziehung gibt. Hier soll entsprechend der Wert **False** als Funktionswert zurückgegeben werden.

### Erweiterung auf Abfragen

Die eingangs geschilderte Aufgabenstellung ist teilweise erledigt, allerdings kann es im Falle von Haupt- und Unterberichten (oder auch Haupt- und Unterformularen) auch einmal vorkommen, dass Abfragen statt Tabellen verwendet werden. Bei Abfragen basierend auf einer Tabelle, die nur zum Beschränken der abzufragenden Felder oder zum Sortieren der Daten dient, können wir die oben beschriebene Funktion verwenden. Wenn die Abfragen jedoch mehr als eine Tabelle verwenden, wird es etwas aufwendiger, Beziehungen zwischen den Datensatzquellen von Haupt- und Unterbericht zu ermitteln.

Hier müssen wir die Funktion also so erweitern, dass diese alle Tabellen der jeweiligen Abfrage untersucht und mit

```
Public Sub Test_GetReference()
    Dim strParenttable As String
    Dim strParentfield As String
    Dim strChildtable As String
    Dim strChildfield As String
    If GetReference("tblArtikel", "tblBestelldetails", strParenttable, strParentfield, strChildtable, strChildfield) _
        = True Then
        If Not (strParenttable = "tblArtikel" And strParentfield = "ArtikelID" _
            And strChildtable = "tblBestelldetails" And strChildfield = "ArtikelID") Then
            Debug.Print "Test 1a fehlgeschlagen 1"
        End If
    Else
        Debug.Print "Test 1a: fehlgeschlagen 2"
    End If
    If GetReference("tblBestelldetails", "tblArtikel", strParenttable, strParentfield, strChildtable, strChildfield) _
        = True Then
        If Not (strParenttable = "tblArtikel" And strParentfield = "ArtikelID" _
            And strChildtable = "tblBestelldetails" And strChildfield = "ArtikelID") Then
            Debug.Print "Test 1b fehlgeschlagen 1"
        End If
    Else
        Debug.Print "Test 1b: fehlgeschlagen 2"
    End If
    If GetReference("tblArtikel", "tblPersonal", strParenttable, strParentfield, strChildtable, strChildfield) _
        = True Then
        Debug.Print "Test 5 fehlgeschlagen"
    End If
End Sub
```

**Listing 2:** Testen der Funktion zum Ermitteln von Beziehungsdaten zweier Tabellen

den Tabellen der anderen Abfrage abgleicht, um Beziehungen zu finden.

### Anzahl der Tabellen einer Datensatzquelle ermitteln

Für die folgenden Schritte wollen wir Tabellen und Abfragen gleich behandeln. In diesen Schritten wollen wir also herausfinden, ob es sich um eine einzelne Tabellen oder eine Abfrage mit einer oder mehreren Tabellen handelt. Den Entwurf einer Tabelle können wir mit dem der **TableDef**-Klasse untersuchen, den einer Abfrage mit der **QueryDef**-Klasse. Wie aber können wir Tabellen und Abfragen auf gleiche Weise untersuchen? Mit dem **Recordset**-Objekt. Dieses untersuchen wir in einer Funk-

tion namens **GetTables** (siehe Listing 3). Die Funktion hat keine andere Aufgabe als ein Array aller in einer Abfrage enthaltenen Tabellen zurückzuliefern. Deshalb verwendet sie neben dem Eingangsparameter **rst** den Ausgangsparameter **strTables** als **String**-Array.

In einer **For Each**-Schleife durchläuft die Funktion alle **Field**-Elemente des **Recordset**-Objekts **rst**. Dabei speichert sie als Erstes den Namen der Tabelle, zu dem das Feld gehört, in der Variablen **strTable**. Diese Quell-tabelle ermitteln wir mit der Eigenschaft **SourceTable** des **Field**-Objekts. Dann versuchen wir, die aktuelle Anzahl der Tabellen im Array **strTables** zu ermitteln, und zwar durch Subtraktion der Ergebnisse der Funktionen

## Ja/Nein-Daten selektieren mit zwei Listenfeldern

Wenn Sie Daten in einer Tabelle haben, die nach einem bestimmten Kriterium in zwei Kategorien eingeteilt werden, benötigen Sie eine praktische Lösung, um die Einteilung vorzunehmen. In diesem Beitrag zeigen wir, wie Sie die Daten einer Tabelle, die beispielsweise über ein Ja/Nein-Feld in zwei Kategorien landen, nach diesem Feld verwalten können. Dazu nutzen wir einfach zwei Listenfelder, von denen eines die Ja-Einträge und eines die Nein-Einträge anzeigt. Das Ändern des relevanten Ja/Nein-Feldes und somit das Verschieben von der einen in die andere Gruppe und umgekehrt erledigen wir dabei per Doppelklick auf das betroffene Element.

Normalerweise erscheinen zwei Listenfelder, welche die Daten der gleichen Tabelle anzeigen, im Kontext von m:n-Beziehungen. Beispielsweise bei Fahrzeugen und Ausstattungsmerkmalen findet sich dann ein Auswahl-feld, mit dem man das Fahrzeug auswählen kann und zwei Listenfelder, von denen das eine die enthaltenen Ausstattungsmerkmale anzeigt und das andere die nicht enthaltenen Ausstattungsmerkmale.

Per Doppelklick oder auch mit entsprechenden Schaltflächen kann der Benutzer dann die Ausstattungsmerkmale hinzufügen oder entfernen.

Das hier zur Unterscheidung zweier Kategorien verwendete **Ja/Nein**-Feld macht es noch einfacher – es gibt keine m:n-Beziehung, sondern nur die Daten einer einzigen Tabelle, von denen die einen mit dem Wert **Ja** und die anderen mit dem Wert **Nein** gekennzeichnet sind.

In unserem Beispiel wollen wir eine Tabelle namens **tblArtikel** verwenden, die ein Feld namens **Auslaufartikel** enthält.

In der Datenblattansicht hat der Benutzer die Gelegenheit, einen Artikel durch das Ändern des Wertes eines Kontrollkästchens als Auslaufartikel zu markieren oder diesen wieder ins Sortiment aufzunehmen.

In der hier vorgestellten Lösung soll es einfacher sein: Wir wollen in einem Listenfeld alle Artikel anzeigen, die nicht auslaufen und in einem zweiten Listenfeld alle Auslaufartikel.

### Beispieltabelle tblArtikel

In diesem Beitrag wollen wir die Tabelle **tblArtikel** der Südsturm-Beispieldatenbank als Beispiel verwenden. Diese stellt mit **Auslaufartikel** ein passendes Feld bereit (siehe Bild 1). Im Bild wird auch gleich ersichtlich, dass

ArtikelID	Artikelname	Auslaufartikel
1	Chai	<input type="checkbox"/>
2	Chang	<input type="checkbox"/>
3	Aniseed Syrup	<input type="checkbox"/>
4	Chef Anton's Cajun Seasoning	<input type="checkbox"/>
5	Chef Anton's Gumbo Mix	<input checked="" type="checkbox"/>
6	Grandma's Boysenberry Spread	<input type="checkbox"/>
7	Uncle Bob's Organic Dried Pears	<input type="checkbox"/>
8	Northwoods Cranberry Sauce	<input type="checkbox"/>
9	Mishi Kobe Niku	<input checked="" type="checkbox"/>
10	Ikura	<input type="checkbox"/>
11	Queso Cabrales	<input type="checkbox"/>
12	Queso Manchego La Pastora	<input type="checkbox"/>
13	Konbu	<input type="checkbox"/>
14	Tofu	<input type="checkbox"/>
15	Genen Shouyu	<input type="checkbox"/>
16	Pavlova	<input type="checkbox"/>
17	Alice Mutton	<input checked="" type="checkbox"/>
18	Carnarvon Tigers	<input type="checkbox"/>

Bild 1: Tabelle mit Auslaufartikeln

die Darstellung in der Datenblattansicht nicht optimal ist: Hier muss man alle Datensätze durchsehen, um die Auslaufartikel zu erkennen. Man könnte auch nach dem Feld **Auslaufartikel sortieren** oder nach dem Wert **Ja** oder **Nein** filtern, aber die Übersicht könnte doch besser sein. Daher erstellen wir nun ein Formular, das die Auslaufartikel und die Nicht-Auslaufartikel in zwei Listenfeldern gegenüberstellt.

### Abfragen zur Auswahl von Auslaufartikel und Nicht-Auslaufartikeln

Bevor wir das Formular programmieren, wollen wir noch die beiden Abfragen erstellen, die als Datensatzherkunft für die beiden Listenfelder dienen. Die erste heißt **qryKeinAuslaufartikel** und verwendet die Tabelle **tblArtikel** als Datenquelle. Sie liefert die drei Felder **ArtikelID**, **Artikelname** und **Auslaufartikel**, wobei das Feld **Auslaufartikel** als Kriterium verwendet wird – die Abfrage soll nur solche Datensätze liefern, deren Artikel keine Auslaufartikel sind (siehe Bild 2). Außerdem sortiert die Abfrage die Artikel aufsteigend nach dem Feld **Artikelname**.

Die zweite Abfrage **qryAuslaufartikel** ist identisch aufgebaut, verwendet aber den Wert **Wahr** als Vergleichswert für das Kriterium mit dem Feld **Auslaufartikel** (siehe Bild 3).

Damit können wir in die Programmierung des Formulars mit den Listenfeldern einsteigen.

### Listenfelder zum Selektieren von Auslaufartikeln

Das neue Formular **frmAuslaufartikelVerwalten** stattdessen wir mit den beiden Listenfeldern **IstKeineAuslaufartikel** und **IstAuslaufartikel** aus.

Für das Formular selbst stellen wir die Eigenschaften **Datensatzmarkierer**, **Navigationsschaltflächen**, **Bildlaufleisten** und **Trennlinien** auf den Wert **Nein** und **Automatisch zentrieren** auf **Ja** ein.

Das Listenfeld **IstKeineAuslaufartikel** verwendet die Abfrage **qryKeinAuslaufartikel** als Datensatzquelle. Da-

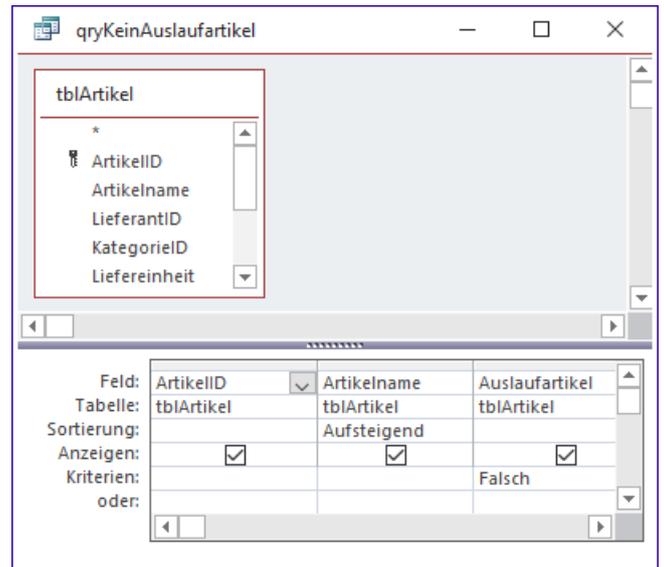


Bild 2: Abfrage mit Artikeln, die nicht ausgelaufen sind

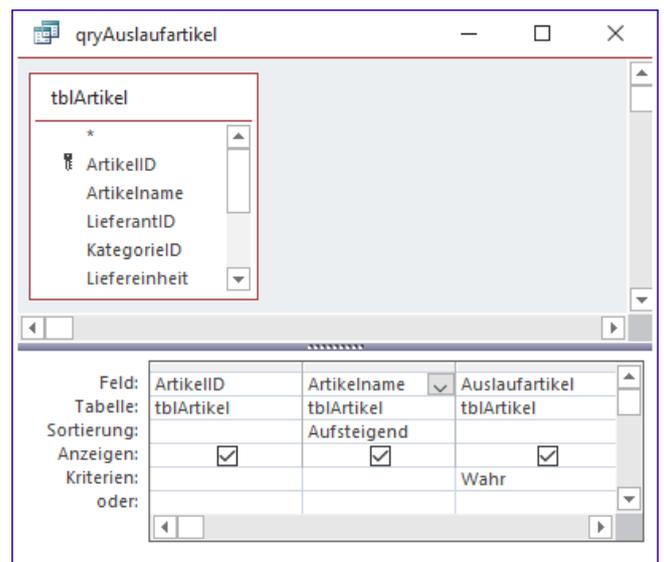


Bild 3: Abfrage mit Artikeln, die ausgelaufen sind

mit nur der Artikelname angezeigt wird, nicht jedoch der Primärschlüsselwert und der Inhalt des Feldes **Auslaufartikel**, stellen wir die Eigenschaft **Spaltenanzahl** des Listenfeldes auf **2** und die Eigenschaft **Spaltenbreiten** auf **0cm** ein.

Damit werden nur die ersten beiden Spalten in die Darstellung aufgenommen und die erste Spalte durch die Spaltenbreite **0cm** praktisch ausgeblendet.

Für das zweite Listenfeld **IstAuslaufartikel** stellen wir die gleichen Eigenschaften ein – mit Ausnahme der Eigenschaft **Datensatzherkunft**, der wir die Abfrage **qryAuslaufartikel** zuweisen.

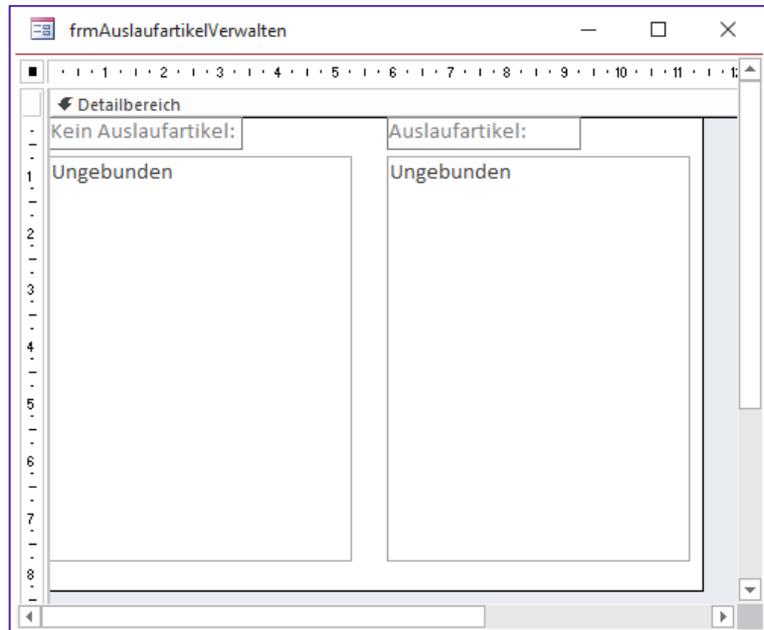
Der Entwurf des Formulars erscheint danach wie in Bild 4.

### Doppelklick-Ereignisse mit Funktion füllen

Damit der Benutzer die Einträge des linken Listenfeldes in das rechte Listenfeld verschieben kann und umgekehrt, sind folgende Schritte nötig:

- Der Wert des Feldes **Auslaufartikel** muss beim Doppelklick auf ein Element auf den umgekehrten Wert eingestellt werden, bei **Wahr** also auf **Falsch** und bei **Falsch** auf **Wahr**.
- Danach müssen die beiden Listenfelder aktualisiert werden, damit der geänderte Eintrag aus dem einen Listenfeld entfernt und zum anderen hinzugefügt wird.

Für das Ereignis **Beim Doppelklicken** des linken Listenfeldes **IstKeineAuslaufartikel** legen wir die Ereignisprozedur aus Listing 1 an. Die Prozedur prüft im ersten



**Bild 4:** Entwurfsansicht des Formulars **frmAuslaufartikelVerwalten**

Schritt, ob der Benutzer einen Eintrag im Listenfeld angeklickt hat oder ob er in den leeren Bereich des Listenfeldes geklickt hat. Nur im ersten Fall werden die innerhalb der **If...Then**-Bedingung befindlichen Anweisungen ausgeführt.

In diesem Fall liest die Prozedur mit der Eigenschaft **ItemsSelected** den Index des angeklickten Eintrags in die Variable **lngIndex** ein. Mit diesem Wert holt sie mit der Ei-

```
Private Sub IstKeineAuslaufartikel_Db1Click(Cancel As Integer)
    Dim db As DAO.Database
    Dim lngIndex As Long
    Dim lngArtikelID As Long
    Set db = CurrentDb
    If Not Me!IstKeineAuslaufartikel.ItemsSelected.Count = 0 Then
        lngIndex = Me!IstKeineAuslaufartikel.ItemsSelected(0)
        lngArtikelID = Me!IstKeineAuslaufartikel.ItemData(lngIndex)
        db.Execute "UPDATE tblArtikel SET Auslaufartikel = True WHERE ArtikelID = " & lngArtikelID, dbFailOnError
        Me!IstAuslaufartikel.Requery
        Me!IstKeineAuslaufartikel.Requery
    End If
End Sub
```

**Listing 1:** Prozedur, die beim Doppelklick auf das Listenfeld **IstKeineAuslaufartikel** ausgelöst wird

```
Private Sub lstAuslaufartikel_DblClick(Cancel As Integer)
    Dim db As DAO.Database
    Dim lngIndex As Long
    Dim lngArtikelID As Long
    Set db = CurrentDb
    If Not Me!lstAuslaufartikel.ItemsSelected.Count = 0 Then
        lngIndex = Me!lstAuslaufartikel.ItemsSelected(0)
        lngArtikelID = Me!lstAuslaufartikel.ItemData(lngIndex)
        db.Execute "UPDATE tblArtikel SET Auslaufartikel = False WHERE ArtikelID = " & lngArtikelID, dbFailOnError
        Me!lstAuslaufartikel.Requery
        Me!lstKeineAuslaufartikel.Requery
    End If
End Sub
```

**Listing 2:** Prozedur, die beim Doppelklick auf das Listenfeld **lstAuslaufartikel** ausgelöst wird

genschaft **ItemData** den Wert der gebundenen Spalte des angeklickten Eintrags und schreibt diesen in die Variable **lngArtikelID**.

Damit können wir nun eine SQL-Anweisung zusammensetzen, die für den Eintrag der Tabelle **tblArtikel**, dessen Feld **ArtikelID** den Wert aus **lngArtikelID** aufweist, den Wert des Feldes **Auslaufartikel** auf **False** einstellt.

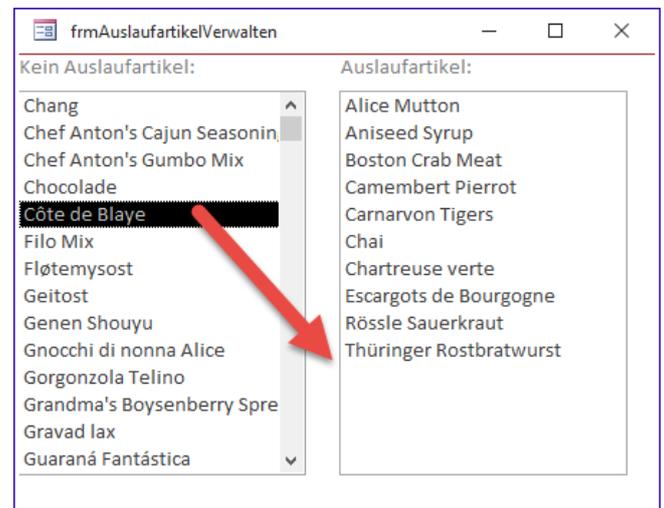
Danach braucht die Prozedur nur noch die beiden Listenfelder mit der **Requery**-Methode zu aktualisieren.

Die Prozedur für das rechte Listenfeld **lstAuslaufartikel** ist ähnlich aufgebaut wie die für das linke Listenfeld (siehe Listing 2). In diesem Fall ermittelt sie den zu bearbeitenden Datensatz allerdings aus dem Listenfeld **lstAuslaufartikel** und ändert den Wert des Feldes **Auslaufartikel** nicht in **False**, sondern in **True**.

### Auslaufartikel verwalten

Damit erhalten wir das Formular aus Bild 5. Wenn Sie auf einen der Einträge im linken Listenfeld klicken, wird dieses in das rechte Listenfeld verschoben und umgekehrt.

Je mehr Datensätze die Tabelle enthält, umso unübersichtlicher wird die Liste. Also lassen wir uns noch etwas einfallen, um die Liste übersichtlicher zu gestalten.



**Bild 5:** Einen Artikel zum Auslaufartikel machen

### Listenfelder schnell filtern

Wie können wir im Listenfeld schnell einen Artikel finden, um diesen als Auslaufartikel zu markieren? Die einfachste Möglichkeit ist, einfach den Anfangsbuchstaben des gewünschten Artikels einzugeben.

Das Listenfeld springt dann automatisch zum ersten Eintrag mit diesem Anfangsbuchstaben.

Manche Listen enthalten aber so viele Einträge, dass auch diese Lösung nicht besonders hilfreich ist. In diesem Fall fügen wir dem Formular ein einfaches Textfeld hinzu, mit

## Ribbon für Rich-Text-Felder

Wenn Sie Daten im Rich-Text-Format anzeigen, können Sie das kleine Popup-Menü zum Formatieren des aktuell markierten Textes nutzen. Manch einem Nutzer ist das lästig und unbequem, da das Menü nur eingeblendet wird, wenn Text markiert ist – und manchmal verschwindet es auch einfach wieder. Da professionelle Anwendungen üblicherweise nicht die eingebauten Ribbon-Einträge anzeigen sollen, stehen wir vor einem Dilemma – wie soll der Benutzer die Formatierungen auf komfortable Weise durchführen? Das ist kein Problem: Wir fügen einfach ein benutzerdefiniertes Ribbon hinzu, das beim Aktivieren eines Textfeldes mit Rich-Text-Format automatisch aktiviert wird und beim Verlassen wieder verschwindet.

Wenn Sie einen Text im Rich-Text-Feld eines Formulars markieren, erscheint das Popup aus Bild 1 mit den Formatierungsmöglichkeiten. Vorher ist es allerdings nicht zu sehen, sodass unerfahrene Benutzer sich wundern, wie Sie Formatierungen für den Text vornehmen sollen. Außerdem wird das Popup-Menü wieder ausgeblendet, wenn sich die Maus nicht mehr über dem markierten und zu formatierenden Bereich befindet.

Wir wollen deshalb ein Ribbon hinzufügen, das nur dann angezeigt wird, wenn der Benutzer das Rich-Text-Feld aktiviert hat. Die Befehle werden dann zunächst deaktiviert angezeigt. Wenn der Benutzer dann Text markiert, werden diese allerdings auch aktiviert.

### Ribbon hinzufügen

Um das Ribbon hinzuzufügen, legen Sie eine Tabelle namens **USysRibbons** an. Diese sieht in der Entwurfsansicht wie in Bild 2 aus. Wenn Sie die Tabelle unter diesem Namen gespeichert haben, werden Sie diese möglicherweise im Navigationsbereich vermissen. Das liegt

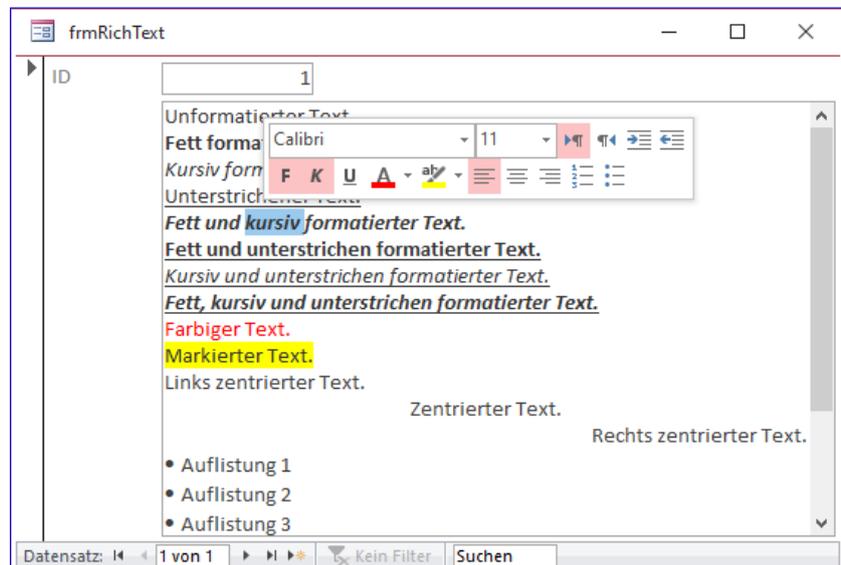


Bild 1: Das etwas unkomfortable Formatierungs-Popup von Rich-Text-Feldern

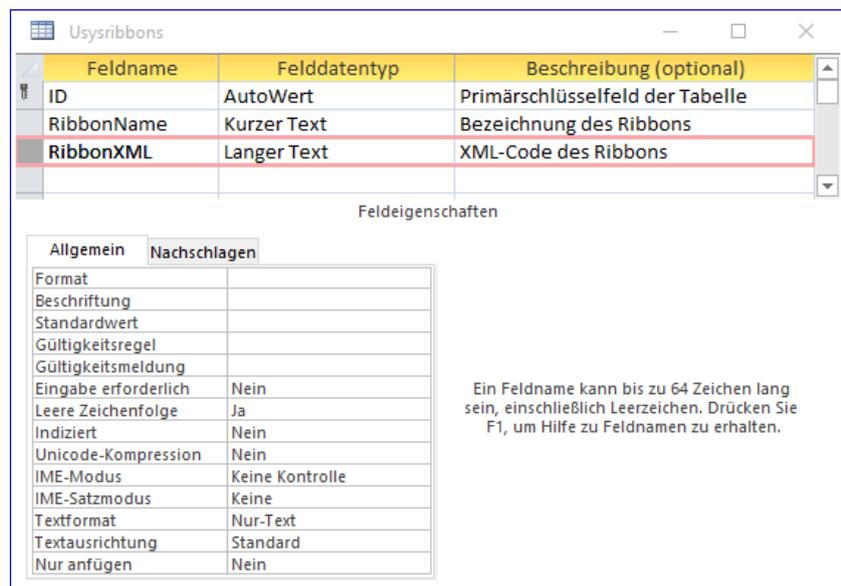


Bild 2: Entwurf der Ribbon-Tabelle

daran, dass der Name die Tabelle als Systemtabelle auszeichnet.

Um sie zu öffnen, aktivieren Sie die Anzeige von Systemobjekten und versteckten Objekten – Sie können diese aber auch einfach über den folgenden Befehl über das Direktfenster des VBA-Editors öffnen:

```
DoCmd.OpenTable "USysRibbons"
```

Danach geben Sie die Ribbon-Definition aus Listing 1 in das Feld **RibbonXML** ein. Dieses Definition legt fest, dass ein neues Ribbon eingeblendet wird und dass alle eingebauten Elemente ausgeblendet werden. Dazu haben wir das Attribut **startFromScratch** auf den Wert **true** eingestellt. Das einzige **tab**-Element des Ribbons enthält ein **Group**-Element, das eine eingebaute Ribbon-Gruppe referenziert. Diese heißt **GroupTextFormatting**.

Diese Gruppe entspricht genau den Elementen, die Sie sehen, wenn Sie das **Start**-Menü von Access aktivieren (siehe Bild 3).

```
<?xml version="1.0"?>
<customUI xmlns="http://schemas.microsoft.com/office/2009/07/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabRichText" label="Formatierung">
        <group idMso="GroupTextFormatting"/>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

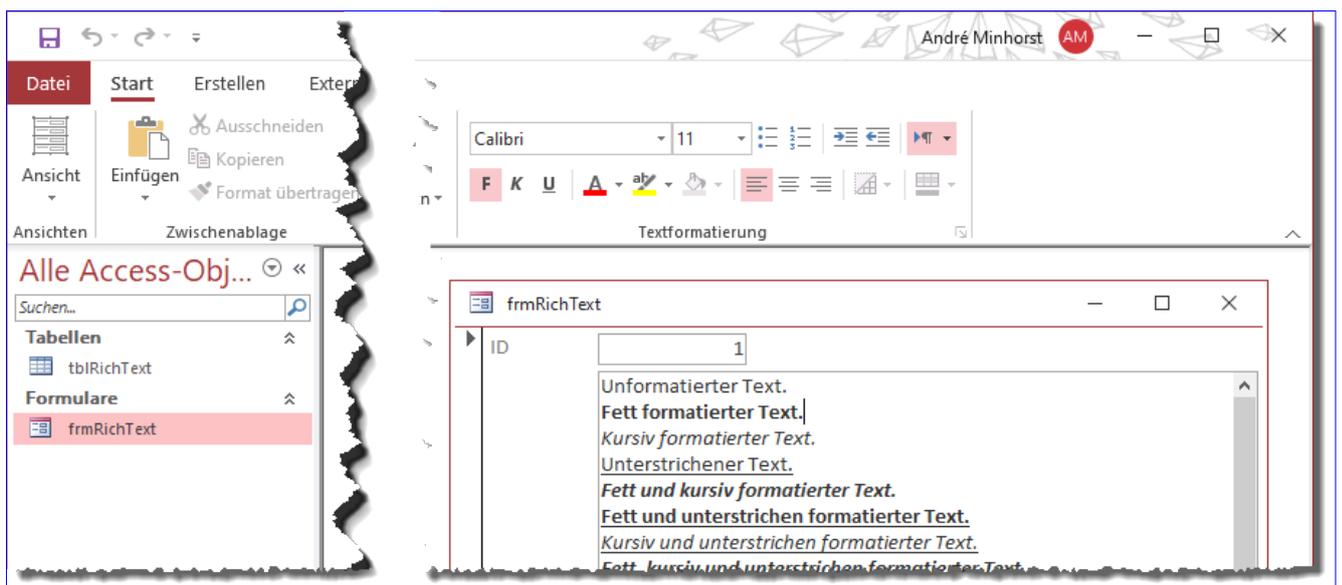
**Listing 1:** Die Definition des Ribbons

Außerdem schreiben wir in das Feld **RibbonName** einen geeigneten Namen, in unserem Fall **RichText**. Diesen Namen benötigen wir gleich noch.

### Name einer eingebauten Gruppe finden

Aber wie finden wir den Namen dieser Gruppe heraus, den wir der Eigenschaft **idMso** des **group**-Elements hinzufügen?

Dazu öffnen wir den Dialog **Access-Optionen** und wechseln zum Bereich **Menüband anpassen**. Hier wählen wir im rechten Bereich den Bereich aus, dessen Namen wir herausfinden möchten – in diesem Fall **StartTextformatierung**. Dann



**Bild 3:** Ribbon-Einträge zum Formatieren des Inhalts von Rich-Text-Feldern

## Unterberichte ein- und ausblenden

Unterberichte bergen oft spannende Herausforderungen in sich – vor allem, wenn die Inhalte sich über mehr als eine Seite erstrecken. Die Aufgabe, die ein Leser mir diesmal stellt, hatte allerdings einen anderen Charakter: Er wünschte sich, dass die Benutzer einzelne Unterberichte je nach Anforderung aktivieren oder deaktivieren konnten. Wir suchen also eine Möglichkeit, um festzulegen, welche von verschiedenen Unterberichten im Hauptbericht angezeigt werden und welche nicht. Wie das gelingt, zeigt der vorliegende Beitrag.

### Unterberichte ein- und ausblenden

Im ersten Schritt wollen wir uns ansehen, ob sich Unterberichte überhaupt ein- und ausblenden lassen. Dazu erstellen wir vier Unterberichte, die wir einfach **rptUnterbericht1**, **rptUnterbericht2**, **rptUnterbericht3** und **rptUnterbericht4** nennen.

Diese sollen jeweils nur einen Detailbereich mit einem Bezeichnungsfeld enthalten, das den Namen des Unterberichts angibt – siehe Beispiel für den Bericht **rptUnterbericht1** in Bild 1.

In einen Hauptbericht namens **rptHauptbericht** fügen wir dann die Unterberichte im Detailbereich ein. Das Ergebnis sieht wie in Bild 2 aus. Damit können wir nun herausfinden, ob sich die Unterberichte überhaupt auf einfache Weise ein- oder ausblenden lassen.

Wenn wir nun in die Seitenansicht wechseln, erscheinen die vier Unterberichte genau so, wie wir sie angeordnet haben (siehe Bild 3).

### Unterberichte ausblenden

Wir beginnen mit dem statischen Umstellen der Sichtbarkeit eines der Unterberichte, indem wir die Eigenschaft

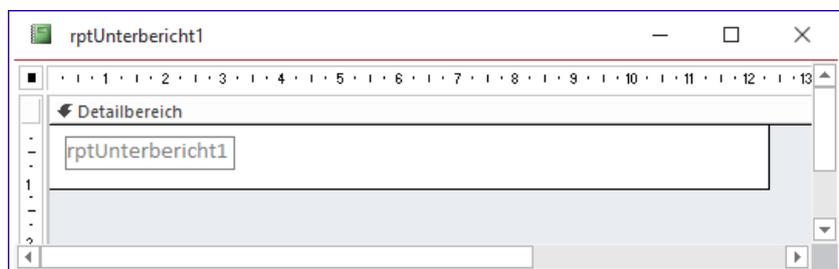


Bild 1: Einer der vier Unterberichte

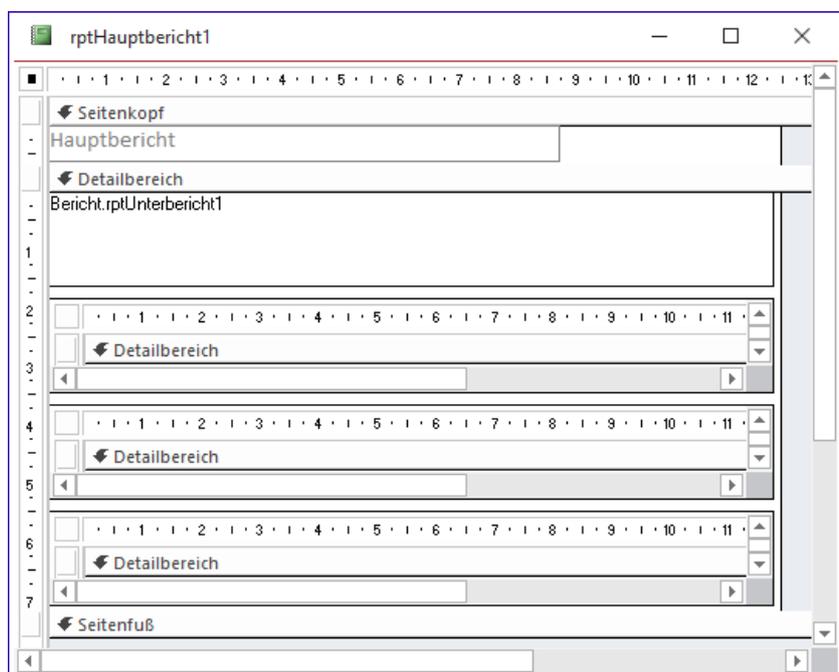
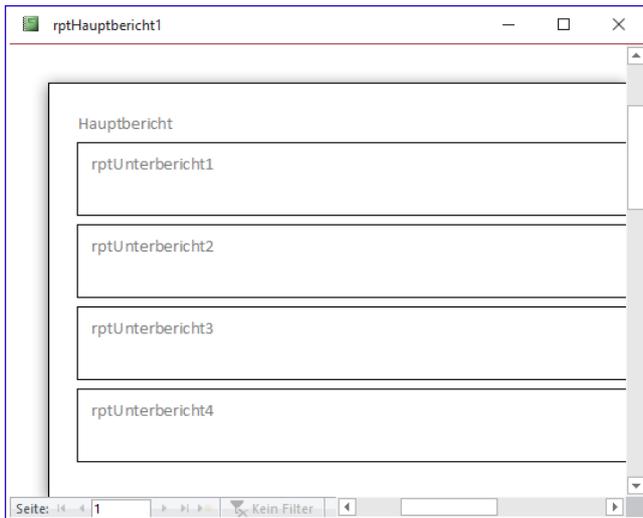
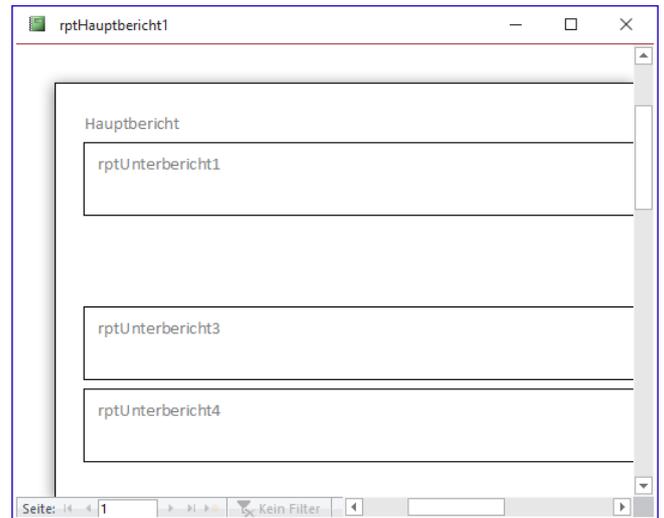


Bild 2: Hauptbericht mit den vier Unterberichten

**Sichtbar** des zweiten Unterbericht-Steuerelements namens **rptUnterbericht2** auf **Nein** einstellen und öffnen den Bericht in der Seitenansicht.



**Bild 3:** Anzeige aller vier Unterberichte



**Bild 4:** Der zweite Unterbericht wurde ausgeblendet.

Das Ergebnis sieht wie in Bild 4 aus und ist noch nicht zufriedenstellend.

Der Unterbericht wird einfach ausgeblendet, aber es bleibt eine leere Stelle, an der sich der ausgeblendete Unterbericht befindet.

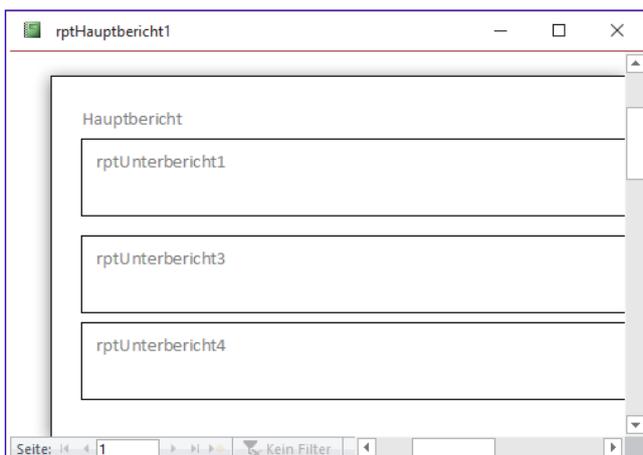
**Leerraum verhindern**

Um den leeren Zwischenraum zwischen dem ersten und dem dritten Unterbericht zu verhindern, stellen wir die Eigenschaft **Verkleinerbar** des Unterbericht-Steuerelements **Untergeordnet2** auf **Ja** ein.

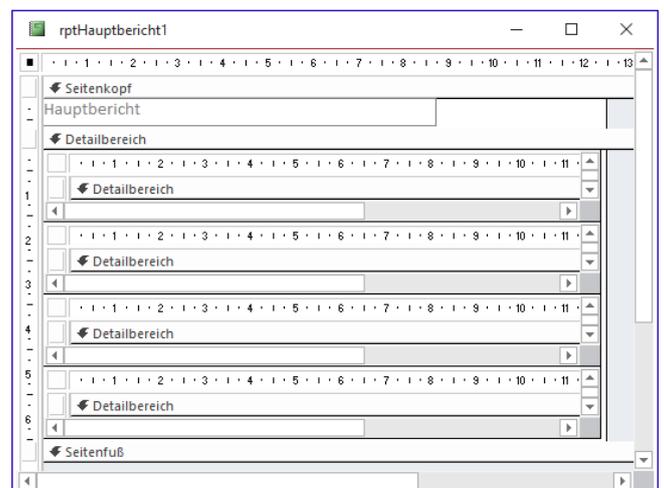
Danach erhalten wir nach dem Wechseln in die Seitenansicht das Resultat aus Bild 5.

Der leere Zwischenraum ist verschwunden und wir finden zwischen dem ersten und dem dritten Unterbericht nur noch die beiden Leerräume zwischen dem ersten und zweiten und dem zweiten und dritten Unterbericht vor.

Das bedeutet wiederum, dass wir, wenn wir beim Ausblenden mehrerer Unterberichte keine Zwischenräume kumulieren wollen, die Unterberichte direkt untereinander anordnen müssen, also etwa so wie in Bild 6.



**Bild 5:** Ausblendete Berichte, die verkleinerbar sind, schaffen Platz für die darunter befindlichen Steuerelemente.



**Bild 6:** Unterberichte ohne Zwischenräume

# SQL Server-Security – Teil 3: SQL Server-Authentifizierung

Bernd Jungbluth, Horn

**Der Zugriff von Access auf die Daten einer SQL Server-Datenbank erfolgt über eine Authentifizierung im SQL Server. Hierfür wird gerne die SQL Server-Anmeldung `sa` verwendet, ist sie doch standardmäßig in jeder SQL Server-Installation vorhanden. So verlockend diese Anmeldung auch sein mag, für einen Zugriffsschutz ist sie nicht empfehlenswert. Der dritte Teil dieser Reihe zeigt Ihnen eine Alternative.**

Die Alternative ist recht einfach: Sie verwenden anstelle der Anmeldung `sa` eine eigene Anmeldung. Doch bevor Sie eine neue Anmeldung im SQL Server anlegen, sollten Sie sich zuerst um die Anmeldung `sa` kümmern.

## Die Anmeldung `sa`

`sa` steht für **System Administration**. Mit der Anmeldung `sa` werden Sie zum Datenbank-Administrator Ihres SQL Servers. Sie dürfen Datenbanken anlegen und löschen, Rechte vergeben und entziehen, alle Daten lesen und ändern und vieles mehr. Kurz gesagt gibt es mit der Anmeldung `sa` für Sie keine Grenzen in Ihrem SQL Server. Das ist auch gut so. Wie sonst könnten Sie den Aufgaben eines Datenbank-Administrators nachkommen? Als solcher sind Sie verantwortlich für die Verfügbarkeit des SQL Servers und der dort enthaltenen Datenbanken. Dies ist nicht nur eine technische Anforderung, sie wird sogar in verschiedenen Gesetzen wie der DSGVO verlangt. Soweit der technische Aspekt. Es geht aber noch weiter. Als Datenbank-Administrator sind Sie schlicht und ergreifend verantwortlich für die Daten. Dies mag zwar nicht zur Definition eines Datenbank-Administrators gehören, aber mal Hand aufs Herz, bei wem steht der Geschäftsführer nach einem Datenverlust oder wenn die Daten durch einen Fremden verfälscht wurden? Richtig, beim Datenbank-Administrator – und der gerät in solchen Situationen schnell in Erklärungsnot.

Sie sollten sich Ihrer Verantwortung als Datenbank-Administrator bewusst sein. Es gehört zu Ihren Aufgaben, die

Daten vor unerlaubtem Zugriff und vor Verlust zu schützen. Ob nun ein Fremder gewollt auf die Daten zugreifen oder ein Angestellter aufgrund zu hoher Rechte die Daten aus Versehen löschen will: Sie müssen entsprechende Maßnahmen treffen, um unerlaubte Zugriffe oder Datenverlust zu verhindern.

Im SQL Server ist das erste Ziel solcher Maßnahmen die Anmeldung `sa`. Geben Sie dieser Anmeldung ein komplexes Kennwort. Dabei gibt es eine einfache Regel zu beachten: Das Kennwort taugt nichts, wenn Sie oder einer Ihrer Kollegen es kennt. Am besten generieren Sie das Kennwort in einem Password-Safe und speichern es dort auch. Nur Sie und die anderen Datenbank-Administratoren haben einen Zugriff auf diesen Password-Safe.

Die Änderung des Kennworts erfolgt im SQL Server Management Studio. In dessen Anmeldedialog wählen Sie als Authentifizierung die SQL Server-Authentifizierung und geben als Anmeldenamen `sa` mitsamt dem zugehörigen Kennwort ein (siehe Bild 1). Nach einem Klick auf **Verbinden** haben Sie Zugriff auf den Objekt-Explorer.

Die folgenden Beispiele beziehen sich auf die früheren Beiträge dieser Reihe. Im Fall der Anmeldung bedeutet dies, dass der »Gemischte Modus« verwendet wird und Sie sich sowohl mit Ihrer Windows-Authentifizierung als auch mit einer SQL Server-Authentifizierung zum SQL Server verbinden können. Sollte dies nicht der Fall sein, beachten Sie bitte die Installationsanleitung der Beispieldateien.

### Hinweis 1: Gemischter Modus – Windows- und SQL Server-Authentifizierung

Im Objekt-Explorer navigieren Sie zum Ordner **Sicherheit** und erweitern den Unterordner **Anmeldungen**. Per Doppelklick auf den Eintrag **sa** öffnen Sie den Dialog **Anmeldungseigenschaften** (siehe Bild 2). Hier geben Sie das neue Kennwort in den Eingabefeldern **Kennwort** und **Kennwort bestätigen** ein. Die zusätzliche Eingabe des alten Kennworts ist für Sie als Datenbank-Administrator nicht erforderlich. Das alte Kennwort müssen Sie nur angeben, wenn Sie ohne administrative Rechte am SQL Server angemeldet sind und ihr eigenes Kennwort ändern möchten beziehungsweise die Kennwörter von anderen Anmeldungen, für die Sie die entsprechenden Freigaben besitzen.

Aktivieren Sie unbedingt die Option **Kennwortrichtlinie erzwingen**. Nur so ist sichergestellt, dass Ihr neues Kennwort den Windows-Kennwortrichtlinien entspricht. Die Windows-Kennwortrichtlinien verwalten Sie in den lokalen Gruppenrichtlinien (siehe Bild 3) oder aber Ihr IT-Systemadministrator legt diese global für alle Computer in Ihrem Unternehmen fest. Diese Option ist sehr hilfreich. Gerade im aktuellen Fall erinnert sie Sie daran, der Anmeldung **sa** ein komplexes Kennwort zu vergeben.

Die Option **Ablauf des Kennworts erzwingen** orientiert sich ebenfalls an den Windows-Kennwortrichtlinien. Bei aktivierter Option müssen Sie das Kennwort in regelmäßigen Abständen ändern. Der Abstand ergibt sich aus den Angaben zum Kennwortalter in den Windows-Kennwortrichtlinien.

Ist es nicht übertrieben, ein generiertes Kennwort, das zudem noch in einem

Password-Safe verwahrt wird, regelmäßig zu ändern? Natürlich nicht! Sollte wider Erwarten jemand Unbefugtes das Kennwort wissen, kann er es nur bis zur nächsten Neuvergabe nutzen. Danach muss er sich wieder die Mühe machen, das Kennwort zu erraten oder es sich per Social Engineering zu erschleichen.

Durch die Aktivierung der Option **Ablauf des Kennworts erzwingen** wird die Option **Benutzer muss das Kennwort bei der nächsten Anmeldung ändern** eingeblendet und aktiviert. Diese Option ist nur interessant, wenn das Kennwort beim ersten Anmeldevorgang durch den Anwender geändert werden soll. In diesem Fall sind Sie

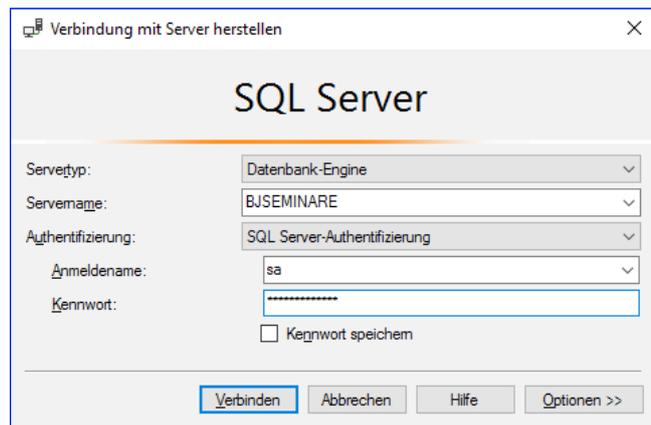


Bild 1: Authentifizierung am SQL Server mit sa

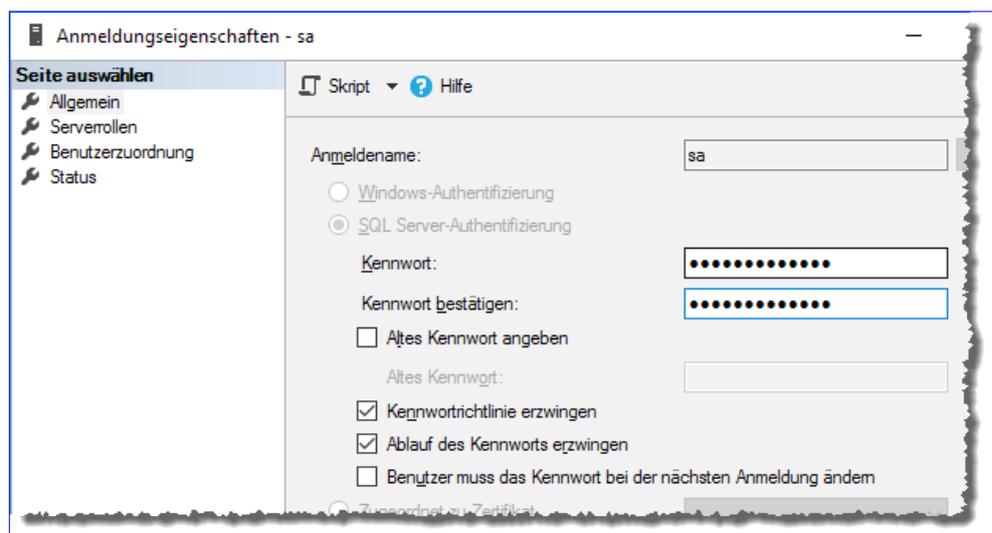


Bild 2: Eigenschaften der Anmeldung sa

selbst der Anwender der Anmeldung und da Sie gerade ein neues Kennwort vergeben, müssen Sie es beim nächsten Anmeldevorgang nicht gleich wieder ändern. Aus diesem Grund können Sie das Häkchen wieder entfernen.

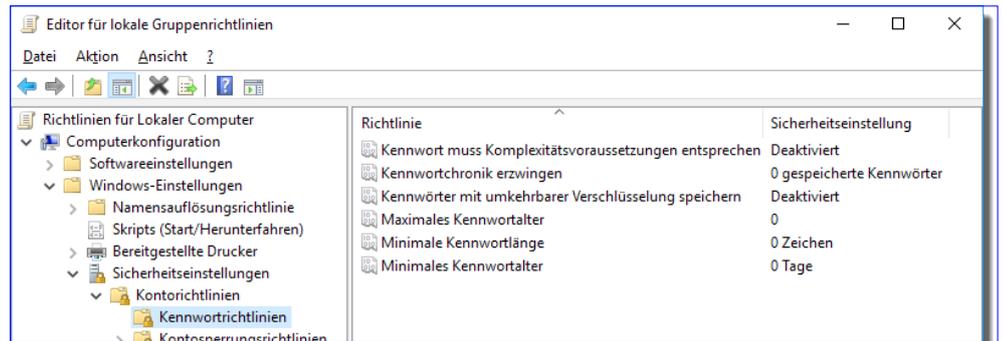


Bild 3: Kennwortvorgaben per Gruppenrichtlinien

Die restlichen Optionen sind für die Neuvergabe des Kennworts nicht relevant. Speichern Sie die Änderung mit einem Klick auf **OK**. Das Kennwort der Anmeldung wurde erfolgreich geändert. Da Sie aktuell diese Anmeldung verwenden, sollten Sie die Verbindung trennen und neu erstellen. Dazu beenden Sie die Verbindung mit der Schaltfläche aus Bild 4.

Anschließend starten Sie den Anmeldedialog direkt wieder mit der Schaltfläche aus Bild 5. Hier wählen Sie erneut die SQL Server-Authentifizierung und geben **sa** als Anmeldenaamen ein. In Ihrem Password-Safe kopieren Sie das Kennwort und fügen es in das Feld **Kennwort** ein. Mit einem Klick auf **Verbinden** melden Sie sich erneut am SQL Server an.

Das war doch einfach. Anmeldedialog starten, **sa** eingeben, Kennwort kopieren und einfügen. Gut, in Zukunft müssen Sie erst noch den Password-Safe öffnen, dort das Kennwort eingeben und innerhalb des Safes den Eintrag **sa** finden. Ist allein der Gedanke daran nicht schon mühsam? Da gibt es eine gute Nachricht. Sie müssen nicht jedes Mal das Kennwort im Password-Safe suchen. Denn die Anmeldung **sa** ist ab jetzt für Sie und alle anderen Datenbank-Administratoren tabu. Nur in absoluten Ausnahme- und Notfällen dürfen Sie es verwenden.

### Anmeldung für Datenbank-Administratoren

Administrative Tätigkeiten erledigen Sie mit Ihrem Windows-Benutzerkon-

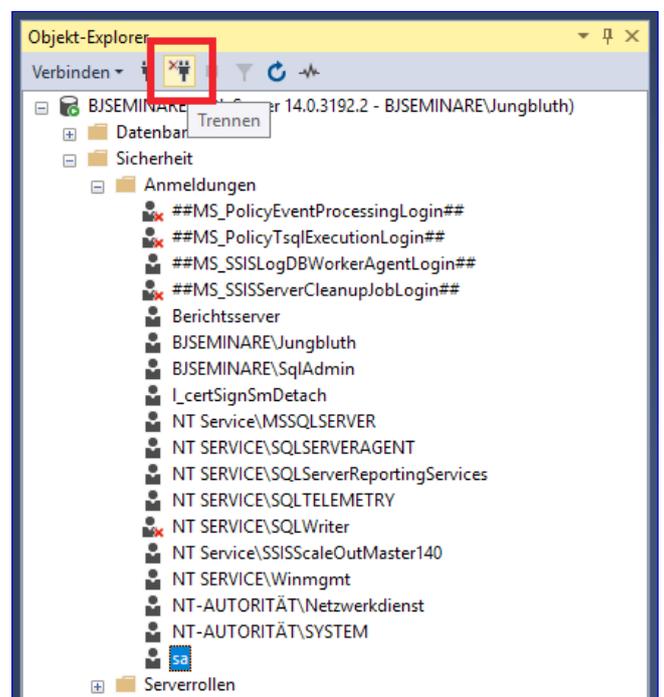


Bild 4: Trennen der Verbindung

to. Die erforderlichen Rechte dazu haben Sie sich vielleicht schon bei der Installation vom SQL Server gegeben. Dort konnten Sie bei der Serverkonfiguration die SQL Server-Administratoren festlegen. Mit einem Klick auf die Schaltfläche

**Aktuellen Benutzer hinzufügen** wurden Sie zum Datenbank-Administrator des SQL Servers und über die Schaltfläche **Hinzufügen** waren Sie in der Lage, weitere Kollegen als Datenbank-Administratoren aufzunehmen (siehe Bild 6).

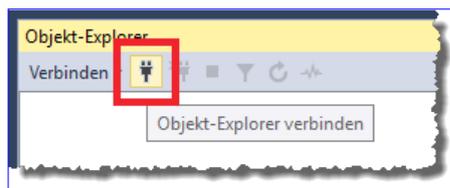


Bild 5: Herstellen der Verbindung

Wenn dem so ist, sehen Sie im Ordner **Anmeldungen** unter **Sicherheit** einen Eintrag mit Ihrem Windows-Benutzerkonto. Diese Anmeldung besitzt die gleichen Rechte wie die Anmeldung **sa**. Sie haben also keinerlei Nachteile, wenn Sie in Zukunft die Anmeldung zu Ihrem Windows-Benutzerkonto verwenden.

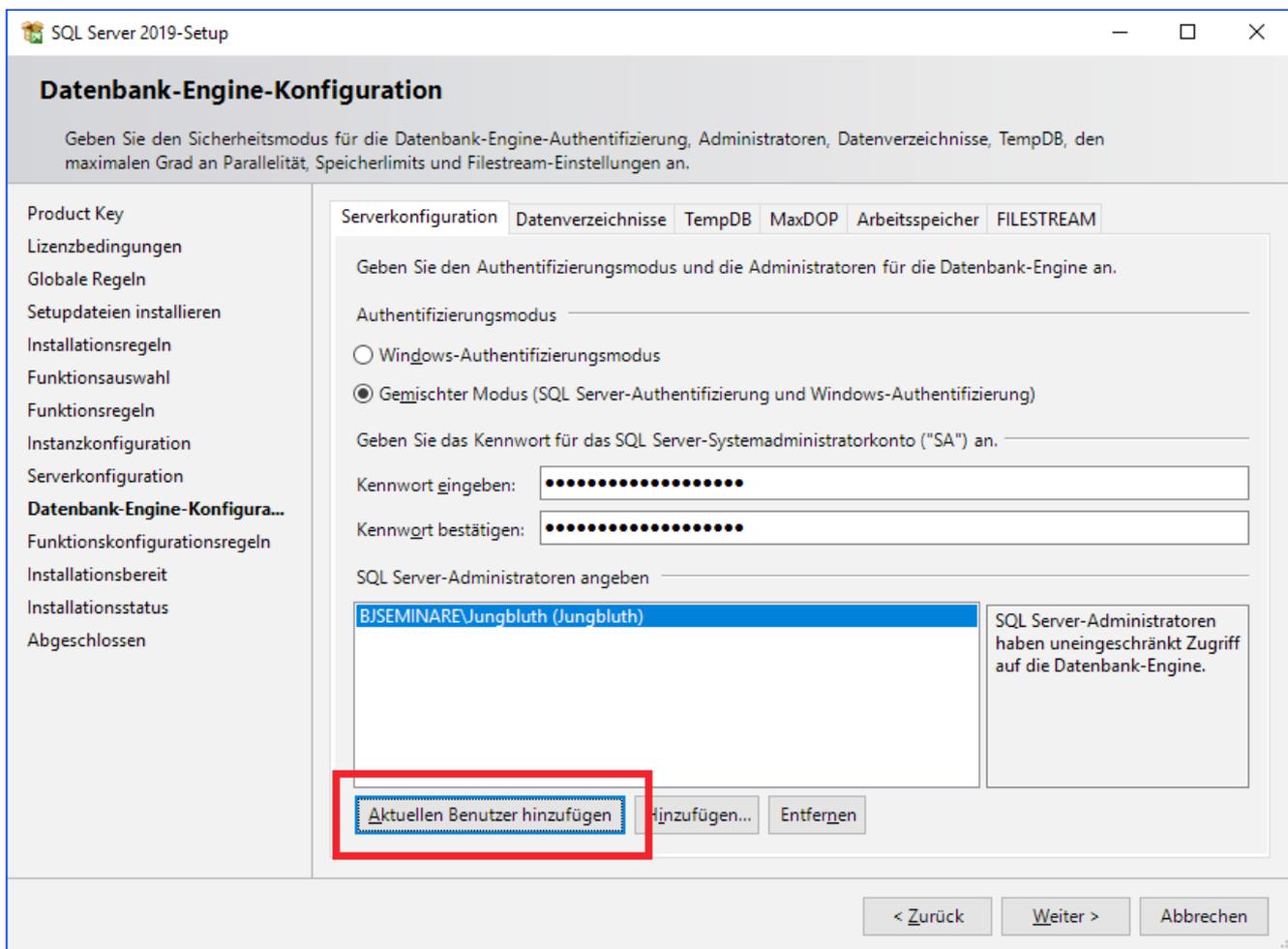
Um wie beim letzten Teil dieser Reihe eine Verwirrung bezüglich der Begriffe **Anwender**, **Benutzer** und **Anmeldung** zu vermeiden, gelten ab hier die wie folgt definierten Begriffe:

- **Anmeldung:** Die Windows- oder SQL Server-Authentifizierung am SQL Server

- **Benutzer:** Der Datenbankbenutzer einer SQL Server-Anmeldung
- **Anwender:** Die Person am Rechner
- **Benutzerkonto:** Das Benutzerkonto der Person am Rechner

Sollten Sie wider Erwarten keine Anmeldung mit Ihrem Benutzerkonto sehen, können Sie dieses jetzt anlegen.

Wählen Sie hierzu im Kontextmenü des Eintrags **Anmeldungen** den Eintrag **Neue Anmeldung**.



**Bild 6:** Definition eines Datenbank-Administrators bei der Installation

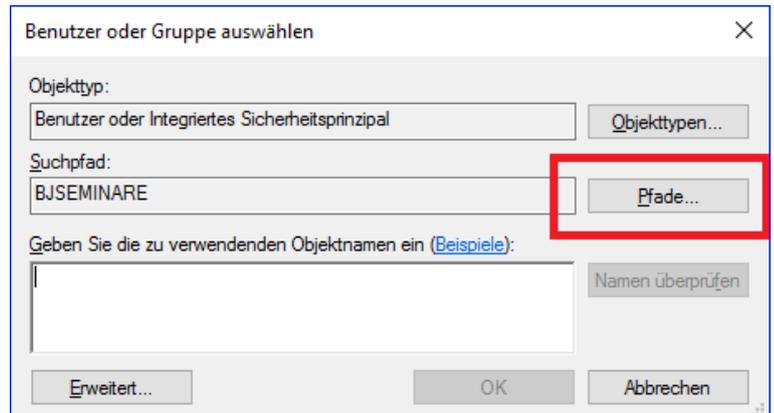
Im Dialog **Anmeldung - Neu** klicken Sie auf **Suchen**. Die Suche ist abhängig vom Suchpfad, den Sie über die Schaltfläche **Pfade** festlegen (siehe Bild 7). Handelt es sich bei Ihrem Benutzerkonto um ein lokales Konto, stellen Sie den Suchpfad auf den Namen des Computers ein. Ist es ein Konto aus Ihrem Active Directory, wählen Sie Ihre Domäne als Suchpfad.

Anschließend tragen Sie Ihren Benutzernamen ein und klicken auf **Namen überprüfen**. Bei einer korrekten Eingabe wird der Benutzername mit dem Namen des Computers oder der Domäne ergänzt. Klicken Sie auf **OK** und wechseln Sie zur Seite **Serverrollen**. Hier markieren Sie den Eintrag **sysadmin** und bestätigen die Auswahl mit **OK**.

Nun existiert eine SQL Server-Anmeldung mit einem Verweis auf Ihr Benutzerkonto. Durch die Zuordnung zur Serverrolle **sysadmin** haben Sie mit dieser Anmeldung die gleichen Rechte wie die Anmeldung **sa**. Es spricht also nichts dagegen, ab jetzt diese Anmeldung zu verwenden.

Die Verbindung zum SQL Server können Sie mit den Ihnen bereits bekannten Schaltflächen trennen und wiederherstellen. Im Anmeldedialog wählen Sie dann bei **Authentifizierung** die Windows-Authentifizierung und klicken auf **Verbinden**. Ein Kennwort müssen Sie dabei nicht eingeben. Die Authentifizierung am SQL Server erfolgt mit Ihrer aktuellen Anmeldung am Betriebssystem. Möglicherweise haben Sie bei der Installation nicht Ihr Benutzerkonto als Datenbank-Administrator angegeben, sondern ein anderes und allgemeineres wie zum Beispiel **SqlAdministrator**. Solche Benutzerkonten können Sie natürlich auch für die Administration des SQL Servers verwenden. Jedoch ist das Anmelden mit einem anderen Benutzerkonto am SQL Server Management Studio nicht so einfach.

Der Anmeldedialog bietet diese Möglichkeit schlicht und ergreifend nicht an. Es bleibt Ihnen nichts anderes übrig, als das gewünschte Benutzerkonto bereits beim Start



**Bild 7:** Suche eines Windows-Benutzerkontos

des SQL Server Management Studios zu verwenden. Dazu melden Sie sich entweder mit dem Benutzerkonto an Ihrem Computer an oder Sie starten das SQL Server Management Studio über die Option **Als anderer Benutzer ausführen**. Dies funktioniert am besten über die Verknüpfung zum SQL Server Management Studio. Klicken Sie die Verknüpfung bei gedrückter Umschalt-Taste mit der rechten Maustaste an und wählen Sie dort den Eintrag **Als anderer Benutzer ausführen**. In dem folgenden Dialog geben Sie den Benutzernamen und das zugehörige Kennwort ein.

Das SQL Server Management Studio stellt nun die Verbindung zum SQL Server mit den Rechten des angegebenen Benutzerkontos her. Diese Möglichkeit zeigt einmal mehr, warum Anwender mit hohen Rechten sehr vorsichtig mit Ihren Kennwörtern umgehen sollten. Dies gilt ebenso für Kennwörter von Sammelkonten wie **SqlAdministrator**.

Grundsätzlich sollten Sie auf Sammelkonten wie **SqlAdministrator** verzichten. Wer ist nochmal verantwortlich für die Daten? Richtig, die Datenbank-Administratoren. Arbeiten alle mit ein und demselben Benutzerkonto, lässt sich nicht nachvollziehen, wer welche administrativen Tätigkeiten im SQL Server durchgeführt hat.

Waren Sie es oder einer Ihrer Kollegen oder gar ein Unbekannter, der sich das Kennwort des Sammelkontos erschlichen hat?

Dies gilt insbesondere dann, wenn es sich bei dem Sammelkonto nicht um eine Windows-Authentifizierung, sondern um eine speziell dafür erstellte SQL Server-Authentifizierung handelt. Wobei die Idee an sich schon unsinnig ist. Wo bitte liegt der Vorteil, anstelle der Anmeldung **sa** eine eigene SQL Server-Authentifizierung mit Administrationsrechten zu nutzen? Eine solche Anmeldung bringt eher Nachteile mit sich, denn mit ihr gibt es eine weitere Anmeldung mit administrativen Rechten, um die Sie sich kümmern müssen.

Sammelkonten sind tabu! Wie sieht es mit Windows-Gruppen aus? In Windows können Sie mehrere Benutzerkonten in Gruppen zusammenfassen. Da ist es doch naheliegend, die Benutzerkonten der Datenbank-Administratoren in eine Windows-Gruppe zu packen und im SQL Server für diese Windows-Gruppe eine Anmeldung mit administrativen Rechten anzulegen. Das ist tatsächlich möglich. SQL Server erlaubt Windows-Gruppen als Anmeldungen. Das Erstellen einer solchen Anmeldung ist identisch mit der eines Benutzerkontos, nur dass Sie anstelle des Benutzerkontos die Windows-Gruppe wählen.

Mit einer Anmeldung basierend auf einer Windows-Gruppe lässt sich keine Verbindung zum SQL Server herstellen. Eine solche Anmeldung definiert lediglich die Rechte für die Mitglieder der Windows-Gruppe. Auf diese Weise können sich die Mitglieder der Windows-Gruppe am SQL Server anmelden und dort mit den zugewiesenen Rechten agieren. Dabei arbeiten diese im SQL Server nicht unter dem Namen der Windows-Gruppe, sondern unter dem ihres eigenen Windows-Benutzerkontos.

Das Konzept der Windows-Gruppen als Anmeldungen im SQL Server ist sehr effektiv und wird in einem gesonderten Beitrag ausführlicher beschrieben. Bei Datenbank-Administratoren jedoch sind Windows-Gruppen nicht zu empfehlen.

Die Pflege der Windows-Gruppen gehört zu den Aufgaben des IT-Systemadministrators. Er kann Mitglieder aus der

Gruppe entfernen und neue aufnehmen. Zum Beispiel könnte er der Windows-Gruppe der Datenbank-Administratoren sein eigenes oder besser noch ein komplett neues Benutzerkonto hinzufügen. Wodurch er im SQL Server mit seinem eigenen oder dem speziell dafür angelegten Benutzerkonto uneingeschränkte Rechte hätte. Er wäre somit in der Lage, alle Daten zu lesen und diese auch zu seinen Gunsten zu ändern.

Wer ist nochmal verantwortlich für die Daten? Sie als Datenbank-Administrator! Verwenden Sie als Anmeldung für die Datenbank-Administratoren eine Windows-Gruppe, geben Sie schlicht und ergreifend die Kontrolle ab. Sie haben keinen Einfluss darauf, wer wann und für wie lange Mitglied dieser Windows-Gruppe ist. Für Anmeldungen mit administrativen Rechten sind Windows-Gruppen genauso tabu wie Sammelkonten.

Es gibt eine einfache und klare Empfehlung für Anmeldungen mit administrativen Rechten: Jeder Datenbank-Administrator erhält im SQL Server seine eigene Anmeldung. Diese Anmeldung verweist auf das persönliche Benutzerkonto des Datenbank-Administrators. Heißen Ihre Datenbank-Administratoren **Hesselbach** und **Hoppenstett**, gibt es im SQL Server eine Anmeldung mit dem Verweis zum Benutzerkonto **Hesselbach** und eine weitere mit dem Verweis zum Benutzerkonto **Hoppenstett**.

Diese Sorgfalt hat noch einen weiteren Vorteil. Erfolgt der Zugang über eine Windows-Gruppe, können sich die Datenbank-Administratoren recht einfach vom SQL Server aussperren. Ist ein Datenbank-Administrator über eine Windows-Gruppe mit dem SQL Server verbunden, kann er die Anmeldung zur Windows-Gruppe ohne weiteres löschen. Worauf ihm beim nächsten Verbindungsversuch der Zugang verwehrt wird, denn ohne die Anmeldung der Windows-Gruppe fehlt ihm die Legitimation.

Ist der Datenbank-Administrator über sein persönliches Benutzerkonto angemeldet, kann er weder seine Anmeldung noch die seiner ebenfalls angemeldeten Kollegen lö-

schen. Anmeldungen mit aktiven Verbindungen lassen sich nicht entfernen, Anmeldungen ohne aktuelle Verbindungen aber durchaus. Auf diese Weise wäre es zwar denkbar, dass sich die Datenbank-Administratoren gegenseitig die Anmeldungen löschen, wenn der jeweilige Kollege gerade nicht angemeldet ist. Es bleibt dabei aber immer eine Anmeldung mit administrativen Rechten übrig und über diese lassen sich die gelöschten Anmeldungen wieder neu erstellen.

Der IT-Systemadministrator hingegen kann durchaus alle Datenbank-Administratoren vom SQL Server aussperren. Dazu braucht er nur versehentlich ihre Benutzerkonten im Active Directory zu löschen. Das erneute Anlegen eines der Benutzerkonten wäre keine Lösung, da hierbei eine neue Security Identifier, kurz **SID**, erstellt wird – und eine SQL Server-Anmeldung über die Windows-Authentifizierung basiert genau auf dieser **SID**.

Für den Fall, dass sich keiner der Datenbank-Administratoren mehr mit dem SQL Server verbinden kann, steht eine Art Ersatzschlüssel zur Verfügung: die Anmeldung **sa**.

Das ist aber auch der einzige legitime Anwendungsfall für diese Anmeldung. Sollten sich tatsächlich alle Datenbank-Administratoren vom SQL Server ausgesperrt haben, darf sich einer von ihnen mit der Anmeldung **sa** zum SQL Server verbinden und den Fehler beheben. Sobald die Datenbank-Administratoren wieder ihre eigenen Anmeldungen besitzen, erhält die Anmeldung **sa** ein neues per Password-Safe generiertes Kennwort. Ab dann arbeiten alle Datenbank-Administratoren wieder mit ihren eigenen Anmeldungen.

Eigentlich ist die Anmeldung **sa** nicht einmal für eine solche Ausnahmesituation erforderlich. Sogar für diesen unwahrscheinlichen Fall bietet SQL Server eine Lösung. In einer ruhigen Minute, wenn kein Anwender mehr am SQL Server angemeldet ist, schließen Sie Ihr SQL Server Management Studio. Anschließend öffnen Sie den SQL Server-Konfigurations-Manager, navigieren zum Ordner

**SQL Server-Dienste** und beenden dort den SQL Server-Dienst (siehe Bild 8). Falls Sie nicht mit der SQL Server Express Edition arbeiten, beenden Sie auch den Dienst vom SQL Server-Agent. Sie finden den SQL Server-Konfigurations-Manager in der Programmgruppe des SQL Servers.

Nun öffnen Sie per Doppelklick auf den SQL Server-Dienst dessen **Eigenschaften**-Dialog und wechseln zur Registerkarte **Startparameter**. Unter **Startparameter angeben** tragen Sie **-m** ein und klicken auf **Hinzufügen** (siehe Bild 9). Durch diesen Parameter startet der SQL Server-Dienst im Einzelbenutzermodus. Nach einem Klick auf **Übernehmen** werden Sie darauf hingewiesen, dass die Änderung erst nach einem Neustart des Diensts wirksam wird.

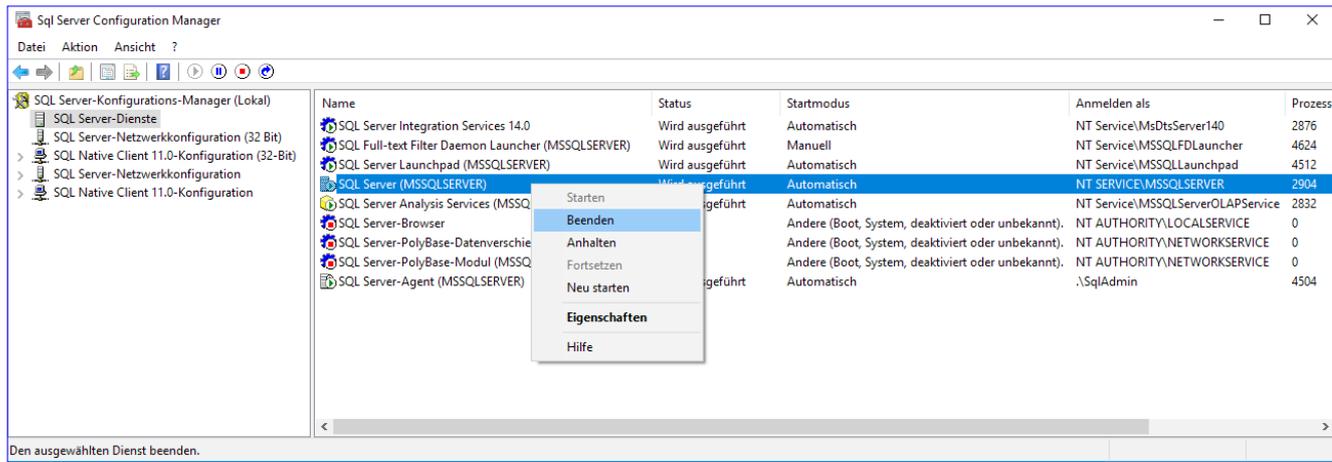
Bestätigen Sie die Meldung und schließen Sie den Dialog mit **OK**. Anschließend wählen Sie im Kontextmenü des SQL Server-Diensts den Eintrag **Starten**.

Nachdem der Dienst neu gestartet ist, klicken Sie mit der rechten Maustaste auf die Verknüpfung zu Ihrem SQL Server Management Studio und wählen **Als Administrator ausführen**. Im Anmeldedialog erstellen Sie die Verbindung zum SQL Server mit der Windows-Authentifizierung. Über diesen Umweg haben Sie nun administrative Rechte im SQL Server und können ungehindert die fehlenden Anmeldungen anlegen.

Sind alle fehlenden Anmeldungen wiederhergestellt, müssen Sie den Einzelbenutzermodus wieder beenden. Dazu schließen Sie das SQL Server Management Studio und wechseln zum SQL Server Konfigurations-Manager. Dort entfernen Sie in den Eigenschaften des SQL Server-Diensts den Eintrag **-m** und starten den Dienst erneut. Denken Sie dabei auch an den Neustart des Diensts vom SQL Server-Agent.

### Der Ersatzschlüssel

Wenn selbst ein solch außergewöhnlicher Fall ohne die Anmeldung **sa** gelöst werden kann, stellt sich die Frage, wofür sie denn überhaupt benötigt wird. Welchen Vorteil



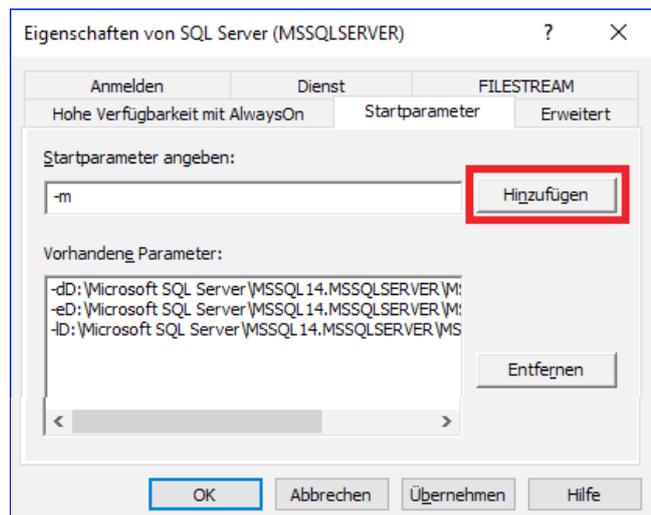
**Bild 8:** Der SQL Server-Dienst im SQL Server Konfigurations-Manager

oder Nutzen bietet die Anmeldung **sa** einem Datenbank-Administrator?

Nicht wenige Datenbank-Administratoren fühlen sich schlicht und ergreifend wohl, wenn ein Ersatzschlüssel in Form der Anmeldung **sa** existiert. Im beschriebenen Fall ist die Lösung mit dem Ersatzschlüssel auch weniger aufwendig. Anstatt erst alle Anwender aufzufordern, die Applikationen mit Zugriff zum SQL Server zu beenden, um dann den SQL Server im Einzelbenutzermodus starten zu können, melden Sie sich einfach per **sa** an und stellen die gelöschten Anmeldungen wieder her.

Auf der anderen Seite ist die Anmeldung **sa** eine nicht unwesentliche Sicherheitslücke. Diese Anmeldung existiert in jedem SQL Server. Möchte sich ein Fremder Zugang zum SQL Server verschaffen, muss er nicht erst mühsam einen Anmeldenamen für seine Attacke ermitteln. Er kann sich darauf verlassen, dass es eine Anmeldung namens **sa** gibt. Das reduziert seinen Aufwand um die Hälfte, weshalb er sich mit aller Energie direkt auf den zweiten Teil konzentrieren wird – das Kennwort.

Zugunsten der Sicherheit sollten Sie auf den Ersatzschlüssel verzichten und die Anmeldung **sa** deaktivieren. Dies ist übrigens die offizielle Empfehlung von Microsoft. Das Deaktivieren findet im **Eigenschaften**-Dialog der An-



**Bild 9:** Die Startparameter des SQL Server-Diensts

meldung **sa** statt. Dort finden Sie in der Seite **Status** die Option **Deaktiviert** (siehe Bild 10). Eine bessere Alternative ist die Deaktivierung mittels der Systemprozedur **sp\_SetAutoSAPasswordAndDisable**. Hierüber erhält die Anmeldung **sa** ein zufällig generiertes Kennwort, bevor sie dann letztendlich deaktiviert wird.

Der Unterschied beider Vorgehensweisen liegt in dem zufällig erzeugten Kennwort. Bei einem einfachen Deaktivieren bleibt das ursprüngliche Kennwort erhalten. Nach einem erneuten Aktivieren der Anmeldung **sa** können Sie direkt wieder mit dem bekannten Kennwort eine Verbindung zum SQL Server herstellen. Dies ist nach einem

# PDF-Dokumente im WebBrowser-Steuerelement

Immer wieder erhalten wir Fragen danach, wie man PDF-Dokumente einfach in Access-Formularen anzeigen kann. Die einfachste Variante scheint die mit dem WebBrowser-Steuerelement zu sein, mit dem man zu einem auf der Festplatte befindlichen PDF-Dokument navigiert. Der vorliegende Beitrag zeigt nicht nur, wie Sie das Dokument anzeigen, sondern liefert auch noch einige Möglichkeiten für die Handhabung des PDF-Dokuments. Dazu gehören das Zoomen, Drucken, Einstellen verschiedener Ansichten und Layouts und vieles mehr.

## Voraussetzungen

Die Lösung zu diesem Beitrag haben wir auf einem Rechner mit Windows 10, Office 365 und Adobe Acrobat Reader DC in der Version 2020.012.20041 entwickelt und getestet.

Wenn wir im **WebBrowser**-Steuerelement die **Navigate2**-Methode verwenden und als Parameter den Pfad zu einem PDF-Dokument angeben, wurde unter den oben genannten Voraussetzungen das angegebene PDF-Dokument innerhalb des Webbrowser-Steuerelements angezeigt – so, als ob Sie vom Internet Explorer aus ein PDF-Dokument öffnen.

## Formular mit Webbrowser-Steuerelement anlegen

Die erste Aufgabe ist das Hinzufügen eines **WebBrowser**-Steuerelements zu einem neuen Access-Formular. Das Formular legen wir in der Beispieldatenbank unter dem Namen **frmPDFImWebbrowser** an.

Diesem fügen wir nun in der Entwurfsansicht ein **WebBrowser**-Steuerelement hinzu, das wir im Ribbon unter **Entwurf | Steuerelemente | Webbrowsersteuerelement** finden.

Nach dem Hinzufügen sieht das Steuerelement recht unscheinbar aus (siehe Bild 1). Für das neue **WebBrowser**-Steuerelement legen wir den Namen **ctlWebbrowser** fest.

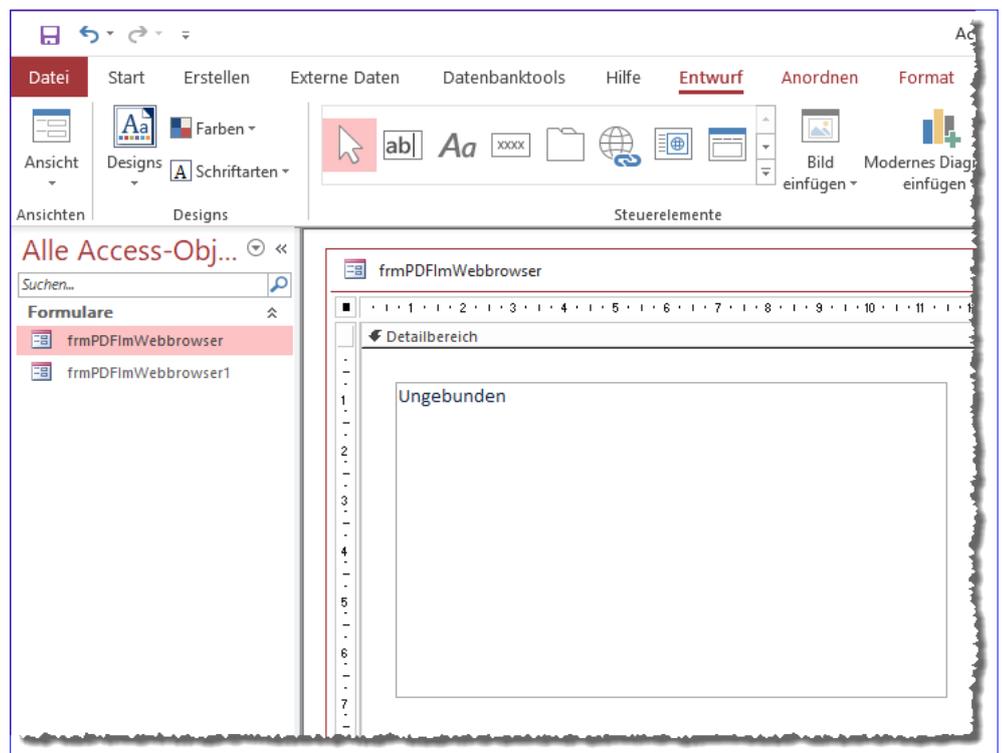


Bild 1: Anlegen des WebBrowser-Steuerelements

Damit sich das **WebBrowser**-Steuerelement beim Vergrößern des Formulars mit vergrößert, stellen wir die beiden Eigenschaften **Horizontaler Anker** und **Vertikaler Anker** auf den Wert **Beide** ein.

Damit keine unnötigen Elemente im Formular angezeigt werden, legen wir außerdem für die Eigenschaften **Datensatzmarkierer**, **Navigationsschaltflächen**, **Bildlaufleisten** und **Trennlinien** den Wert **Nein** fest.

## Schaltfläche zum Öffnen eines PDF-Dokuments

Nun beginnen wir mit den Elementen, die wir für das Öffnen eines PDF-Dokuments im **WebBrowser**-Steuerelement benötigen. Dazu fügen wir eine einfache Schaltfläche namens **cmdPDFOeffnen** hinzu.

Für diese hinterlegen wir die folgende Ereignisprozedur:

```
Private Sub cmdPDFOeffnen_Click()  
    Dim strPDF As String  
    strPDF = CurrentProject.Path & "\\test.pdf"
```

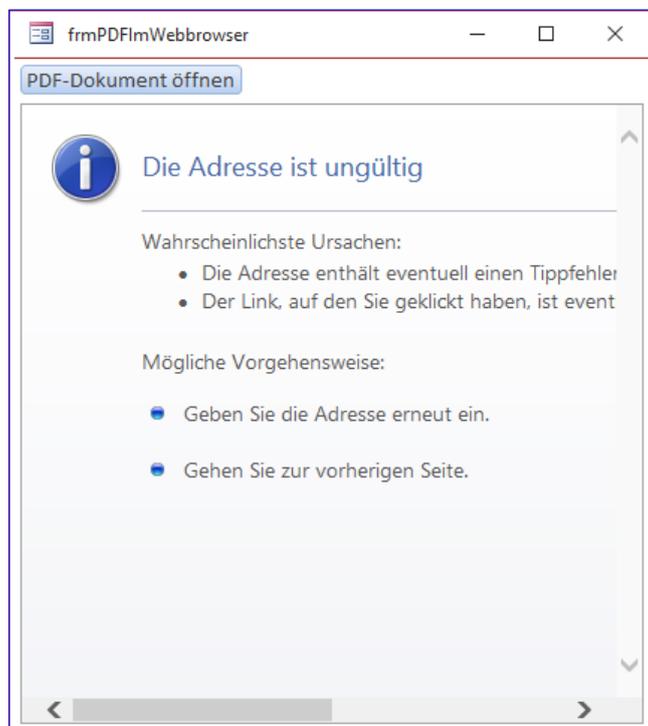


Bild 2: WebBrowser-Steuerelement ohne Inhalt

```
Me!ctlWebbrowser.Object.Navigate strPDF  
End Sub
```

## PDF anzeigen

Der erste Wechsel in die Formularansicht ist ernüchternd: Das **WebBrowser**-Steuerelement zeigt lediglich eine Meldung über eine ungültige Adresse an (siehe Bild 2).

Das ändert sich allerdings nach einem Klick auf die Schaltfläche **cmdPDFOeffnen**. Das **WebBrowser**-Steuerelement zeigt dann wie in Bild 3 das gewünschte PDF-Dokument an.

## Leeres Browserfenster anzeigen

Wir wollen dem Benutzer beim Öffnen des Formulars nicht die Meldung über die ungültige Adresse präsen-

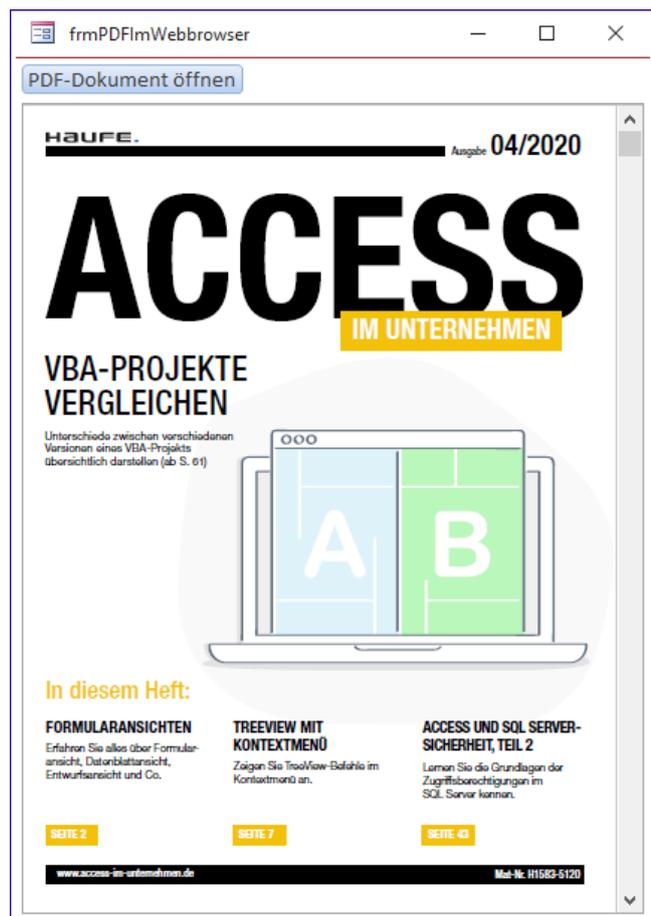


Bild 3: PDF-Dokument im WebBrowser-Steuerelement

tieren, sondern ein leeres Browser-Fenster. Dabei erledigen wir direkt ein paar Vorarbeiten für die folgenden Schritte. Als Erstes fügen wir dem Klassenmodul des Formulars ganz oben die folgende Deklaration einer Variablen für das **Webbrowser**-Element hinzu. Dieses stattdessen wir mit dem Schlüsselwort  **WithEvents**  aus, damit wir seine Ereignisse im Klassenmodul implementieren können. Eines davon benötigen wir nämlich:

```
Dim WithEvents objWebbrowser As WebBrowser
```

Danach implementieren wir eine Ereignisprozedur für das Ereignis  **Beim Laden**  des Formulars. Hier füllen wir direkt die Variable  **objWebBrowser**  mit dem Element, das in der Eigenschaft  **Object**  des  **WebBrowser** -Steuerelements steckt. Der Hintergrund ist, dass das von uns hinzugefügte Access-Steuerelement nur ein Wrapper für das eigentliche  **WebBrowser** -Steuerelement ist, der einige Eigenschaften und Ereignisse zur Verfügung stellt.

Wir benötigen aber auch beispielsweise ein Ereignis, das dieser Wrapper nicht bereithält. Also greifen wir mit  **Me!ctlWebbrowser.Object**  auf das darin enthaltene  **WebBrowser** -Steuerelement zu:

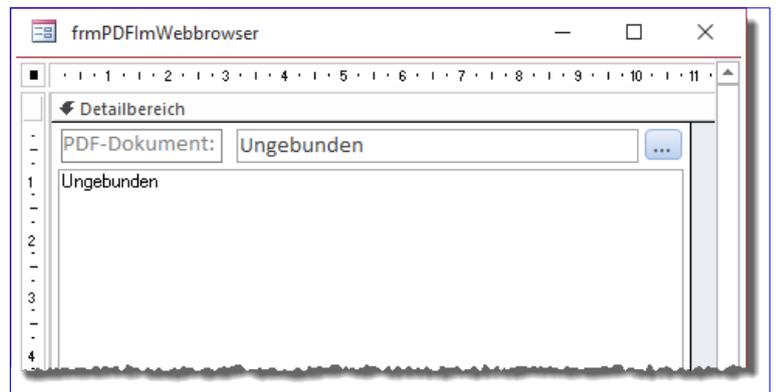
```
Private Sub Form_Load()  
    Set objWebbrowser = Me!ctlWebbrowser.Object
```

Anschließend verwenden wir die  **Navigate2** -Methode, um zu einer Seite namens  **about:blank**  zu navigieren. Danach zeigt das  **WebBrowser** -Steuerelement statt der Seite mit der Meldung eine leere Seite an:

```
objWebbrowser.Navigate2 "about:blank"  
End Sub
```

### PDF auswählen

Das Laden des PDF-Dokuments wollen wir noch ein wenig eleganter gestalten, denn schließlich wollen wir nicht



**Bild 4:** Steuerelemente zum Auswählen des anzuzeigenden PDF-Dokuments

immer das gleiche PDF-Dokument laden, sondern dem Benutzer auch die Möglichkeit geben, dieses auszuwählen.

Dazu fügen wir im oberen Bereich ein Textfeld namens  **txtPDFDocument**  hinzu. Die Schaltfläche  **cmdPDFOeffnen**  stattdessen wir mit drei Punkten als Text aus, sodass der obere Teil des Formulars nun wie in Bild 4 aussieht.

Die Prozedur, die durch diese Schaltfläche ausgelöst wird, ändern wir wie folgt:

```
Private Sub cmdPDFOeffnen_Click()  
    Dim strPDF As String  
    strPDF = OpenFileName(CurrentProject.Path, _  
        "PDF auswählen", "PDF-Dokumente (*.pdf)")  
    Me!txtPDFDokument = strPDF  
    objWebbrowser.Navigate2 strPDF  
End Sub
```

Damit können wir das PDF-Dokument nun mit einem Dateiauswahl-Dialog öffnen.

### Programmierung der PDF-Ansicht

Die PDF-Ansicht des Acrobat-Readers im  **WebBrowser** -Steuerelement bietet einige Möglichkeiten zum Anzeigen von Seiten, Drucken des Dokuments oder zum Einstellen der verschiedenen Ansichten. Hier ist eine Übersicht der nachfolgend vorgestellten Methoden:

- **gotoFirstPage**: Navigiert zur ersten Seite des Dokuments.
- **gotoPreviousPage**: Navigiert zur vorherigen Seite des Dokuments.
- **gotoNextPage**: Navigiert zur nächsten Seite des Dokuments.
- **gotoLastPage**: Navigiert zur letzten Seite des Dokuments.
- **setCurrentPage**: Zeigt die Seite mit der als Parameter angegebenen Seitennummer an.
- **setLayoutMode**: Stellt das Layout auf einen der folgenden Werte ein, den Sie als Parameter bei Aufruf übergeben: **DontCare** verwendet die aktuellen Einstellungen, **SinglePage** zeigt eine einzelne Seite an; **OneColumn** zeigt jeweils eine Seite an; **TwoColumnLeft** zeigt zwei Seiten nebeneinander an, wobei die erste Seite auf der linken Seite beginnt (das heißt, Seite 1 und 2 erscheinen ganz oben); **TwoColumnRight** zeigt ebenfalls zwei Seiten nebeneinander an, aber in der obersten Reihe erscheint nur die erste Seite, und zwar rechts.
- **setPageMode**: Stellt den Seitenmodus auf einen der folgenden Werte ein, den Sie als Parameter beim Aufruf der Methode übergeben: **none** zeigt das Dokument ohne Lesezeichen oder Thumbnails an (Standardeinstellung); **bookmarks** zeigt das Dokument und Lesezeichen an; **thumbs** zeigt das Dokument und Thumbnails an.
- **setView**: Stellt die Ansicht ein und erwartet zum Beispiel einen der folgenden Werte als Parameter: **Fit** passt die Seite so in das Steuerelement ein, dass es sowohl horizontal als auch vertikal hineinpasst; **FitH** passt die Seitenbreite so an, dass die Seite horizontal in das Steuerelement hineinpasst; **FitV** passt die Seitenhöhe so an, dass die Seite vertikal in das Steuerelement passt; **FitB** passt die Seite ebenfalls horizontal und vertikal an das Steuerelement an.
- **setShowScrollbars**: Erwartet einen Boolean-Wert als Parameter, der angibt, ob die Bildlaufleisten angezeigt werden sollen oder nicht.
- **setShowToolbar**: Erwartet einen Boolean-Wert als Parameter, der angibt, ob die Menüleiste angezeigt werden soll oder nicht.
- **LoadFile**: Lädt das als Parameter angegebene PDF-Dokument und zeigt es an. Dies funktioniert allerdings erst, wenn das **WebBrowser**-Steuerelement bereits ein PDF-Dokument anzeigt und die Variable mit dem Verweis auf das PDF-Objekt gefüllt ist.
- **printAll**: Druckt den kompletten Inhalt des Dokuments, ohne einen Drucken-Dialog anzuzeigen.
- **printAllFit**: Druckt den kompletten Inhalt des Dokuments und erwartet einen **Boolean**-Wert als Parameter, der angibt, ob die PDF-Seite in den bedruckbaren Bereich des Druckers eingepasst werden soll.
- **printWithDialog**: Soll einen Drucken-Dialog öffnen, mit dem der Benutzer selbst die Druck-Einstellungen festlegen kann. Der Drucken-Dialog wurde unter unserer Konfiguration allerdings nicht angezeigt.
- **printPages**: Erwartet zwei Parameter, die angeben, welche die erste und welche die letzte zu druckende Seite ist. Bleibt einer dieser Parameter leer, wird die erste beziehungsweise letzte Seite angenommen.
- **printPagesFit**: Erwartet drei Parameter, von denen die ersten beiden angeben, welche die erste und die letzte zu druckende Seite ist. Der dritte Parameter legt fest, ob die PDF-Seite in den bedruckbaren Bereich des Druckers eingepasst werden soll. Bleiben der erste oder

zweite Parameter leer, verwendet die Methode die erste oder letzte Seite des Dokuments als Werte.

- **src**: Eine der wenigen Eigenschaften des Objekts. Hiermit können Sie den Pfad zum aktuell angezeigten PDF-Dokument ermitteln.
- **setZoom**: Erwartet einen Parameter, der angibt, welcher Zoom-Faktor eingestellt werden soll. Der Wert **100** entspricht 100%.
- **setZoomScroll**: Erwartet drei Parameter. Der erste Parameter gibt wie bei **setZoom** den Zoom-Faktor an. Der zweite und dritte Parameter geben an, wie weit vom oberen oder linken Rand nach unten beziehungsweise rechts gescrollt werden soll. Dies hat für uns nur für die vertikale Komponente funktioniert.

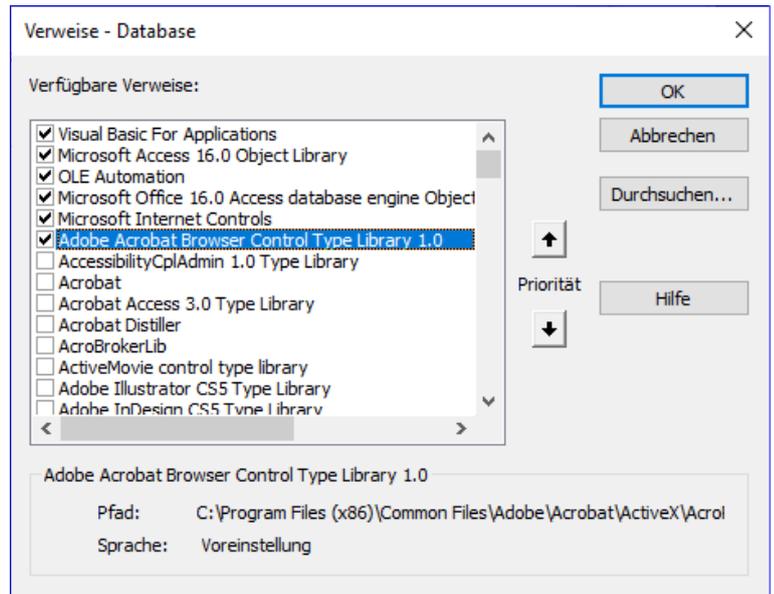
### PDF-Funktionen in das Formular integrieren

Nun fügen wir dem Formular einige Steuerelemente hinzu, mit denen wir die zuvor beschriebenen Methoden auf das aktuell angezeigte PDF-Dokument anwenden können.

Dazu müssen wir allerdings zunächst noch einen Verweis auf die Bibliothek mit den Methoden hinzufügen und das geladene PDF-Dokument mit einer Variablen referenzieren.

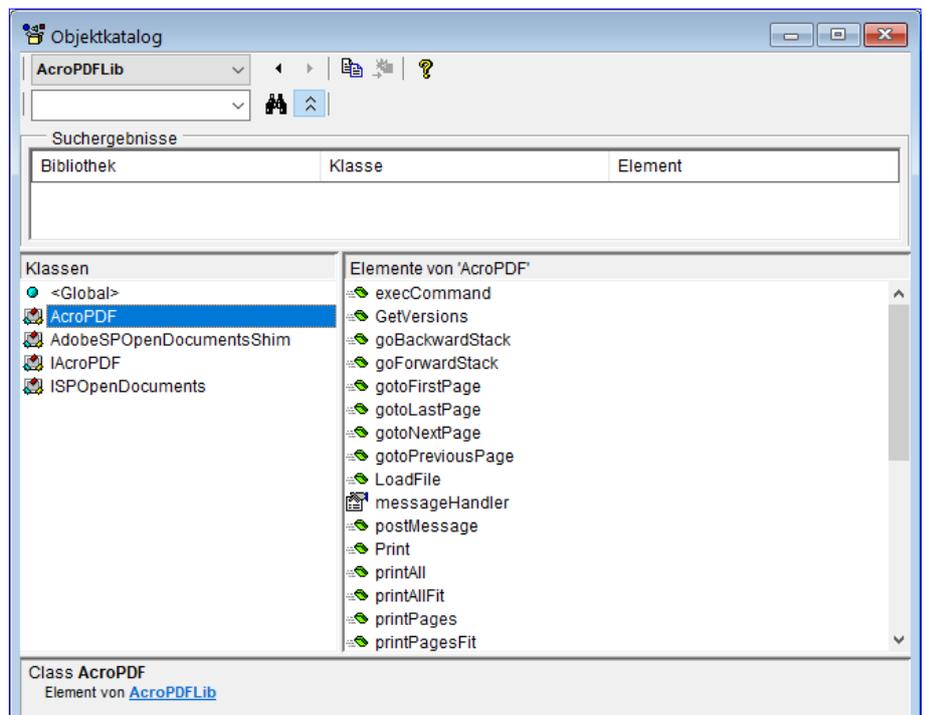
### Verweis auf die Acrobat-Bibliothek

Den Verweis fügen Sie wie üblich über den **Verweise**-Dialog hinzu, den Sie mit dem Menüeintrag **Extras|Verweise** des VBA-Editors



**Bild 5:** Verweis auf die Bibliothek **Adobe Acrobat Browser Control Type Library 1.0**

öffnen. Hier sollten Sie einen Eintrag mit dem Text **Adobe Acrobat Browser Control Type Library 1.0** finden und hinzufügen (siehe Bild 5).



**Bild 6:** Die Klasse **AcroPDF** im Objektkatalog

Anschließend finden Sie im Objektkatalog alle Methoden des Objekts **AcroPDF**. Dazu wählen Sie im Kombinationsfeld oben links den Eintrag **AcroPDFLib** aus und klicken dann in der Liste der Klassen auf den Eintrag **AcroPDF** (siehe Bild 6).

## Objektvariable für das AcroPDF-Objekt

Damit wir die Methoden des **AcroPDF**-Objekt komfortabel nutzen können, deklarieren wir dafür eine passende Objektvariable:

```
Dim objAcrobat As AcroPDFLib.AcroPDF
```

Wann aber füllen wir diese Variable? Wir können das noch nicht in dem Moment erledigen, wo wir das Formular öffnen, denn dann ist noch kein PDF-Dokument im **WebBrowser**-Steuerelement enthalten und somit auch nicht das PDF-Steuerelement.

Wir können die Variable **objAcrobat** erst füllen, wenn der Benutzer das erste PDF-Dokument geladen hat. Wir wissen zwar, wann der Ladevorgang des PDF-Vorgangs

beginnt – nämlich nachdem das **WebBrowser**-Steuerelement die **Navigate2**-Methode ausgeführt hat:

```
objWebbrowser.Navigate2 strPDF
```

Hier kommt nun das Schlüsselwort **WithEvents** für die Variable **objWebBrowser** zum Zuge: Dadurch können wir nämlich die Ereignisse dieses Steuerelements im aktuellen Klassenmodul implementieren. Dazu wählen wir im VBA-Editor im linken Kombinationsfeld den Eintrag **objWebbrowser** und im rechten Kombinationsfeld den Eintrag **NavigateComplete2** aus. Der VBA-Editor legt dann automatisch eine Prozedur namens **objWebbrowser\_NavigateComplete2** an (siehe Bild 7).

In unserem Fall füllen wir die Ereignisprozedur mit den folgenden Anweisungen:

```
Private Sub objWebbrowser_NavigateComplete2(  
    ByVal pDisp As Object, URL As Variant)  
    On Error Resume Next  
    Set objAcrobat = Me!ctlWebbrowser.Object.Document
```

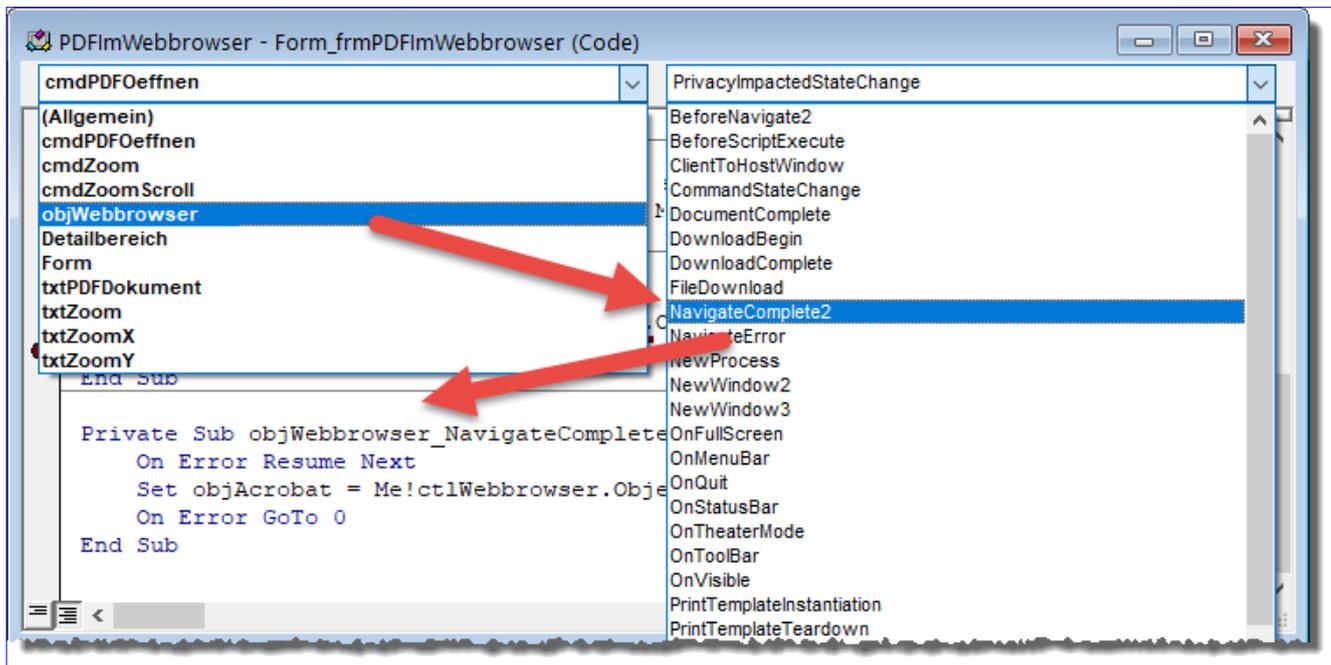


Bild 7: Anlegen einer Ereignisprozedur für das **WebBrowser**-Steuerelement