

Hier den Kundennamen auswählen:

Zum Erstellen der einfachen Rechnung hier klicken!

Stopp

## Inhalt

Office Integration .....	2
Benutzervorlage .....	2
Datenquelle .....	4
Zuordnung .....	4
Projekt-Explorer .....	5
Ereignisprozeduren .....	6
Ereignis Document_Open .....	6
Ereignis cboShipName_Change .....	7
Ereignis cmdStart_Click .....	7
Ereignis cmdStopp_Click .....	7
Prozeduren in Standardmodulen .....	8
Modul mdlGeneral .....	8
Modul mdlFillCboShipName .....	8
Modul mdlFillInvoiceForm .....	8
Modul mdlOtherProcs .....	13
Ergebnis .....	15
Ergänzungen .....	16

## Abbildungen

Abbildung 1: Einfaches Rechnungsformular mit elf Textmarken und 3 Tabellen .....	3
Abbildung 2: Elf Textmarken für das einfache Rechnungsformular .....	3
Abbildung 3: Beziehungen der ACCESS-Tabellen in der Abfrage <i>Invoices</i> .....	4
Abbildung 4: Ebenen des Projekt-Explorers in WORD .....	5
Abbildung 5: Automatisch ausgefülltes Rechnungsformular für Alfreds Futterkiste .....	15
Abbildung 6: <i>Dokument1</i> vor dem Speichern im Projekt-Explorer .....	15

## Listings

Listing 1: Eine öffentliche Variable im ThisDocument-Klassenmodul deklarieren .....	6
Listing 2: Ereignisprozedur Document_Open .....	6
Listing 3: Ereignisprozedur bei Änderung des Kombinationsfeldes .....	7

Listing 4: Aktionen der Befehlsschaltfläche cmdStart .....	7
Listing 5: Anwendung beenden .....	7
Listing 6: Datei-öffentliche Konstanten und Variablen .....	8
Listing 7: Kombinationsfeld mit eindeutigen Kundennamen füllen .....	9
Listing 8: Rechnungsformular füllen (1. Teil) .....	10
Listing 9: Rechnungsformular füllen (2. Teil) .....	10
Listing 10: Rechnungsformular füllen (3. Teil) .....	11
Listing 11: Rechnungsformular füllen (4. Teil) .....	12
Listing 12: Rechnungsformular füllen (5. Teil) .....	12
Listing 13: Rechnungsformular füllen (6. Teil) .....	13

## Tabellen

Tabelle 2: Zuordnung von Feldnamen und Textmarken .....	5
Tabelle 3: Aufgabenbeschreibung der Standardmodule .....	6
Tabelle 4: Übersicht über aufgerufene und aufrufende Prozeduren .....	13

## Office Integration

In diesem Beitrag wird beschrieben, wie mit Microsoft Office WORD eine benutzerdefinierte Dokumentvorlage mit Rechnungsdaten aus der ACCESS-Beispieldatenbank Northwind.mdb gefüllt werden kann.

Die programmgestützte Lösung dafür benötigt drei sog. ActiveX-Steuerelemente<sup>1</sup> in diesem Dokument, siehe oben:

- Ein Kombinationsfeld zur Auswahl des Kundennamens durch den Benutzer.
- Ein Befehlsschaltfläche zum Start der Rechnungserstellung nachdem ein Kundename ausgewählt wurde.
- Eine Befehlsschaltfläche zum Beenden der Anwendung.

Die Anwendung setzt voraus, dass ...

- eine benutzerdefinierte Dokumentvorlage mit dem Namen *Simple\_Invoice.dotx* im Standardpfad für Benutzervorlagen gespeichert ist,
- die MS ACCESS Beispieldatenbank *Northwind.mdb* auf dem lokalen Rechner implementiert ist,
- eine Prozedur für das Ereignis *Document\_Open* vorhanden ist, die das erwähnte Kombinationsfeld beim Öffnen des Dokuments automatisch mit Einträgen belegt.

## Benutzervorlage

Die benötigte benutzerdefinierte Dokumentvorlage sei bereits erstellt und unter dem Namen *Simple\_Invoice.dotx* im Standardpfad für WORD-Benutzervorlagen gespeichert. Das Layout der Benutzervorlage ist aus Abbildung 1 ersichtlich. Sie enthält 3 Tabellen:

- Die erste Tabelle enthält 5 geschlossene Textmarken für Kundendaten

---

<sup>1</sup> Die ActiveX-Steuerelemente in WORD 2010 sind die aus WORD 2003 bekannten Steuerelemente der Steuerelemente-Toolbox.

## Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen

Dr. V. Thormählen

- Die zweite Tabelle enthält 6 geschlossene Textmarken für den Rechnungsempfänger
- Die dritte Tabelle ist noch leer. Sie dient später zur Aufnahme von beliebig vielen Rechnungspositionen mit den beiden Datenelementen *Produkt* und *€-Betrag*.

**Einfaches Rechnungsformular**

**Kunde**

CoName
CoAddress
CoCity
CoState
CoZip

**Rechnungsempfänger**

BillToName
BillToCompany
BillToAddress
BillToCity
BillToState
BillToZip

**Rechnung**


Abbildung 1: Einfaches Rechnungsformular mit elf Textmarken und 3 Tabellen

Die Namen der 11 geschlossenen Textmarken in der Benutzervorlage können mit dem Navigationspfad Register *Einfügen* ▶ Bereich *Hyperlinks* ▶ Schaltfläche *Textmarke* angezeigt werden, siehe Abbildung 2.

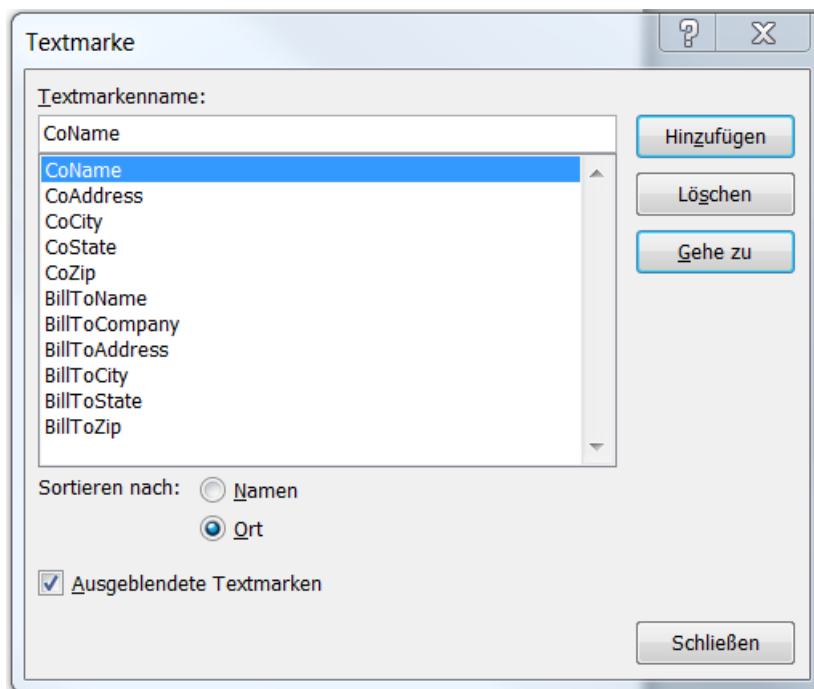


Abbildung 2: Elf Textmarken für das einfache Rechnungsformular

**Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen**  
 Dr. V. Thormählen

Benutzervorlagen werden in WORD am sinnvollsten mit Textmarken (engl. bookmarks) gesteuert. Es gibt 2 Arten:

- offene Textmarken
- geschlossene Textmarken (sog. Textmarkenfelder)

Der Unterschied liegt im markierten Textbereich: Ist ein Textbereich markiert, umfasst die Textmarke diesen Bereich, anderenfalls fügt WORD lediglich eine offene Textmarke an der Stelle der Schreibmarke ein.

Bei den erwähnten 11 Textmarken in der Benutzervorlage *Simple\_Invoice.dotx* handelt sich um *geschlossene* Textmarken.

Mit dem Navigationspfad *Datei* ▶ *Optionen* ▶ *Anzeige* ▶ *Textmarken anzeigen* lassen sich in WORD 2010 Textmarken anzeigen bzw. verbergen.

**Datenquelle**

Als Datenquelle zum Füllen des einfachen Rechnungsformulars wird die Abfrage *Invoices* der MS ACCESS-Beispieldatenbank *Northwind.mdb* benutzt, siehe Abbildung 3.

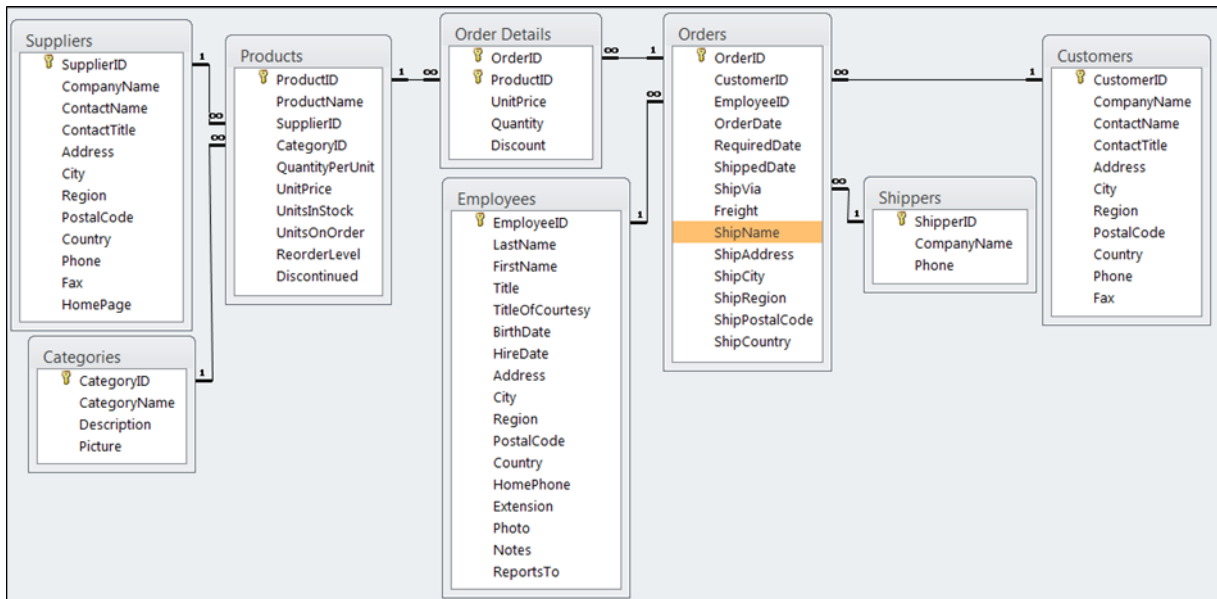


Abbildung 3: Beziehungen der ACCESS-Tabellen in der Abfrage *Invoices*

**Zuordnung**

Tabelle 1 beinhaltet die Zuordnungen zwischen den Datensätzen aus der Abfrage *Invoices* und den geschlossenen Textmarken der Benutzervorlage *Simple\_Invoice.dotx*, siehe Abbildung 1.

Feldnamen der Datensätze aus MS ACCESS	Textmarken in der Dokumentvorlage
ShipName	BillToCompany
ShipAddress	BillToAddress
ShipCity	BillToCity
ShipRegion	BillToState
ShipPostalCode	BillToZip
ShipCountry	
CustomerID	
Customers.CompanyName	CoName

Address	CoAddress
City	CoCity
Region	CoState
PostalCode	CoZip
Country	
Salesperson	BillToName
OrderID	
OrderDate	
RequiredDate	
ShippedDate	
Shippers.CompanyName	
ProductID	
<b>ProductName</b>	keine (Für Spalte „Produkt“ in der 3. Tabelle)
UnitPrice	
Quantity	
Discount	
<b>ExtendedPrice</b>	keine (Für Spalte „Betrag“ in der 3. Tabelle)
Freight	

Tabelle 1: Zuordnung von Feldnamen und Textmarken

## Projekt-Explorer

Die weitere Beschreibung orientiert sich an der Hierarchie des *Projekt-Explorers*, die aus mehreren Ebenen besteht, siehe Abbildung 4.

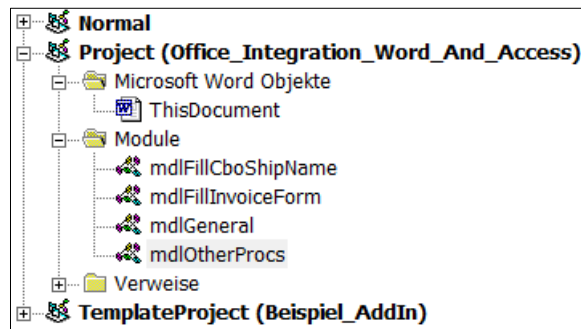


Abbildung 4: Ebenen des Projekt-Explorers in WORD

- Die oberste Ebene beinhaltet das aktuelle Projekt: *Office\_Integration\_Word\_And\_Access*.
- *ThisDocument* bezeichnet das Klassenmodul für das aktuelle Dokument.
- Die unterste Ebene zeigt die einzelnen Module an, hier sind es 4 Standardmodule.

Hinter dem Eintrag *ThisDocument* verbirgt sich ein sog. *Klassenmodul*<sup>2</sup>. Es enthält Ereignisprozeduren. Diese Prozeduren werden von WORD unter bestimmten Bedingungen automatisch aufgerufen ohne dass der Benutzer zwingend etwas davon bemerkt. Die Prozedur *Document\_Open* zählt beispielsweise dazu, siehe Listing 2.

Der Eintrag *Module* enthält vier Standardmodule. Sie dienen zur Strukturierung des VBA<sup>3</sup>-Codes im Projekt, siehe Tabelle 2.

<sup>2</sup> WORD unterscheidet zwei Grundtypen von Modulen: *Klassenmodule* und *Standardmodule*.

<sup>3</sup> VBA ist das Kürzel für **V**isual **B**asic for **A**pplications

Name des Moduls	Aufgabe des Moduls
mdlGeneral	Deklariert Datei-öffentliche Konstanten und Variablen
mdlFillCboShipName	Füllt das Kombinationsfeld <i>cboShipName</i> mit eindeutigen Kundennamen aus der ACCESS-Beispieldatenbank <i>Northwind.mdb</i>
mdlFillInvoiceForm	Füllt ein einfaches Rechnungsformular mit Daten aus der ACCESS-Beispieldatenbank <i>Northwind.mdb</i> unter Berücksichtigung eines vom Benutzer ausgewählten Kundennamens.
mdlOtherProcs	Enthält 6 Prozeduren der zweiten Aufrufebene. Diese aufgeteilt in 1 Sub-Prozedur und 5 Function-Prozeduren.

Tabelle 2: Aufgabenbeschreibung der 4 Standardmodule

Im Folgenden wird der Quellcode (engl. source code) aller Module gezeigt. Kommentare (bzw. Meldungen) sind im Quellcode **gelb** (bzw. **grau**) hervorgehoben.

## Ereignisprozeduren

Das vorliegende WORD-Dokument (*Office\_Integration\_Word\_And\_Access.docm*) enthält zahlreiche Prozeduren.

Beim Öffnen des Dokuments wird der VBA-Code in der Ereignisprozedur *Document\_Open* automatisch ausgeführt, also ohne weiteres Zutun des Benutzers. Hauptzweck dieser Prozedur ist der Aufruf der Subroutine *FillCboShipName*. Diese belegt das Kombinationsfeld (*ThisDocument.cboShipName*) mit Kundennamen, die der Abfrage *Invoices* der ACCESS-Beispieldatenbank *Northwind.mdb* entnommen werden, siehe Listing 2.

```
Option Explicit
' Eine öffentliche Variable im ThisDocument-Klassenmodul deklarieren
Dim m_strShipNm As String
```

Listing 1: Eine öffentliche Variable im ThisDocument-Klassenmodul deklarieren

## Ereignis Document\_Open

```
Private Sub Document_Open()
' Ereignisprozedur beim Öffnen des Dokuments
If Not CheckInlineShapes() Then
MsgBox "Steuerelement(e) fehlen" & vbCrLf & _
"Die Prozedur wird beendet.", vbCritical, "Document_Open"
WordDateiSpeichernUndSchliessen
End If
With ThisDocument
.cboShipName.Enabled = True
.cmdStopp.Enabled = True
End With
If CheckNumberOfBookmarks(11) And CheckNumberOfTables(3) Then
' Diese Prozedur ausführen: Kombinationsfeld füllen
FillCboShipName
Else
MsgBox "Die benutzerdefinierte Dokumentvorlage ist fehlerhaft" & vbCrLf & _
"Die Prozedur wird beendet.", vbCritical, "Document_Open"
WordDateiSpeichernUndSchliessen
End If
End Sub
```

Listing 2: Ereignisprozedur Document\_Open

## Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen

Dr. V. Thormählen

Sobald der Benutzer einen Kundennamen im gefüllten Kombinationsfeld ausgewählt hat, wird die Ereignisprozedur *cboShipName\_Change* automatisch ausgelöst. Dort wird der gewählte Kundename in der öffentlichen Variablen *m\_strShipNm* gespeichert, siehe Listing 1 und Listing 3. Der Gültigkeitsbereich dieser Variablen erstreckt sich auf alle Ereignisprozeduren im Klassenmodul *ThisDocument*. Außerdem wird die Befehlsschaltfläche *cmdStart* zum Starten der Rechnungserstellung aktiviert.

### Ereignis *cboShipName\_Change*

```
Private Sub cboShipName_Change()  
    ' Den gewählten Eintrag des Kombinationsfelds cboShipName  
    ' in der öffentliche Variablen m_strShipNm speichern  
    m_strShipNm = ThisDocument.cboShipName.Value  
    ThisDocument.cmdStart.Enabled = True  
End Sub
```

Listing 3: Ereignisprozedur bei Änderung des Kombinationsfeldes *cboShipName*

Die Befehlsschaltfläche *cmdStart* löst folgendes aus, siehe Listing 4:

- Wenn ein Apostroph im Kundennamen vorkommt, wird dieser verdoppelt. Das erfordert die Syntax für SQL-Abfragen in MS ACCESS
- Die Prozedur (*FillInvoiceForm*) zur Rechnungserstellung mit dem gewählten Kundennamen wird ausgeführt.
- Die Befehlsschaltfläche *cmdStart* wird wieder deaktiviert.

### Ereignis *cmdStart\_Click*

```
Private Sub cmdStart_Click()  
    ' Rechnungserstellung starten  
    If InStr(1, m_strShipNm, "'") > 0 Then  
        m_strShipNm = Replace(m_strShipNm, "'", "'")  
    End If  
    ' Diese Prozedur ausführen: Rechnungsformular ausfüllen  
    Call FillInvoiceForm(m_strShipNm)  
    ThisDocument.cmdStart.Enabled = False  
End Sub
```

Listing 4: Aktionen der Befehlsschaltfläche *cmdStart*

Die Befehlsschaltfläche *cmdStop\_Click* beendet die Anwendung und löst die Verbindung zur Beispieldatenbank *Northwind.mdb*, siehe Listing 5.

### Ereignis *cmdStopp\_Click*

```
Private Sub cmdStopp_Click()  
    ' Anwendung beenden  
    With ThisDocument  
        With .cboShipName  
            .Clear  
            .Enabled = False  
        End With  
        .cmdStart.Enabled = False  
        .cmdStopp.Enabled = False  
    End With  
    WordDateiSpeichernUndSchliessen  
    ' Verbindung zu Northwind.mdb schließen  
    p_cnn.Close  
    Set p_cnn = Nothing  
End Sub
```

Listing 5: Anwendung beenden

## Prozeduren in Standardmodulen

### Modul mdlGeneral

Dieses Standardmodul dient zur expliziten Deklaration von Datei-öffentlicher Konstanten und Variablen. Die hier deklarierten Konstanten und Variablen sind gültig in allen Modulen, also innerhalb des gesamten Word-Dokuments mit dem Namen *Office\_Integration\_Word\_And\_Access*, siehe Listing 6.

```
Option Explicit
' Deklaration von Datei-öffentlichen Konstanten
Public Const p_cstrTemplate As String = "Simple_Invoice.dotx"
Public Const p_cstrDBPath As String =
"D:\Access\MS_Download\Northwind\Northwind.mdb"
' Deklaration von Datei-öffentlichen Variablen
Public p_cnn As New ADODB.Connection ' Verbindung zu Northwind.mdb
Public p_strConnect As String ' Verbindungszeichenfolge
Public p_strSQL As String ' SQL-Befehl
Public p_intRow As Integer ' Zeilenzähler
```

Listing 6: Datei-öffentliche Konstanten und Variablen

Alle Konstanten und Variablen sind mit dem jeweils geeigneten Datentyp deklariert. Zur besseren Lesbarkeit werden die Präfixe der entsprechenden Namen klein geschrieben. Der eigentliche Name beginnt mit einem Großbuchstaben und zwischendrin ggf. noch weiteren Großbuchstaben.

### Modul mdlFillCboShipName

Dieses Standardmodul enthält nur die Sub-Prozedur *FillCboShipName*. Drei lokale Variablen werden explizit im Kopf der Sub-Prozedur deklariert. Auf eine ausführliche Ablaufbeschreibung wird verzichtet, denn sie enthält zahlreiche Kommentare, siehe Listing 7.

### Modul mdlFillInvoiceForm

Dieses Standardmodul enthält nur die Sub-Prozedur *FillInvoiceForm*. Sie dient dazu, relevante Daten aus der Beispieldatenbank *Northwind.mdb* zu entnehmen und in das einfache Rechnungsformular einzufügen. Beim ihrem Aufruf muss ein gültiger Kundename in der Variablen *strShipNm* als Zeichenkette übergeben werden, siehe 1. Codezeile in Listing 8. Danach erfolgt wiederum die Deklaration der lokalen Variablen.



Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen  
Dr. V. Thormählen

```
Public Sub FillCboShipName ()
    ' Verweis: Microsoft ActiveX Data Objects 6.1 Library
    ' Aufgabe: Kombinationsfeld 'cboShipName' mit eindeutigen Kundennamen füllen
    ' Deklaration der lokalen Variablen
    Dim rstCust As New ADODB.Recordset ' Datensätze für Kundennamen
    Dim intCust As Integer ' Zahl der Kundennamen
    Dim varCust As Variant ' Datenbereich für Kundennamen
    ' Fehlerbehandlung definieren
    On Error GoTo Err_Point
    ' Verbindungszeichenfolge zur Northwind.mdb bestimmen
    p_strConnect = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Persist Security Info=False;" & _
        "Data Source=" & p_cstrDBPath
    p_cnn.Open p_strConnect
    ' SQL-Befehl zur Abfrage eindeutiger Kundennamen (ShipName)
    ' aus der Abfrage Invoices
    p_strSQL = "SELECT DISTINCT ShipName " & _
        "FROM Invoices " & _
        "ORDER BY ShipName"
    ' Datenbereich f. Kundennamen erstellen und SQL-Abfrage ausführen
    Set rstCust = New ADODB.Recordset
    rstCust.Open Source:=p_strSQL, ActiveConnection:=p_strConnect, _
        CursorType:=adOpenStatic, LockType:=adLockReadOnly
    ' Zahl der eindeutigen Kundennamen ermitteln
    intCust = rstCust.RecordCount
    If intCust = 0 Then
        MsgBox "Keine Kundennamen gefunden", vbCritical, "FillCboShipName"
        GoTo Exit_Point
    End If
    ' Feld vom Typ Variant (varCust) mit dem Inhalt des Feldes
    ' 'ShipName' der Abfrage 'Invoices' in Nothwind.mdb füllen
    varCust = rstCust.GetRows(Rows:=intCust)
    ' Kombinationsfeld (cboShipName) mit eindeutigen Kundennamen füllen
    With ThisDocument.cboShipName
        .Clear
        For p_intRow = 0 To intCust - 1
            ' Eintrag hinzufügen
            .AddItem varCust(0, p_intRow)
        Next p_intRow
        ' Keinen Eintrag auswählen
        .ListIndex = -1
    End With
    ' Befehlsschaltfläche cmdStart sperren
    ThisDocument.cmdStart.Enabled = False
Exit_Point:
    On Error Resume Next
    ' Datensätze für Kundennamen schließen
    rstCust.Close
    Set rstCust = Nothing
    Exit Sub
Err_Point:
    MsgBox "Fehler #: " & Err.Number & ", " & Err.Description, vbCritical,
    "Fill_Cbo_ShipName"
    Resume Exit_Point
End Sub
```

Listing 7: Kombinationsfeld mit eindeutigen Kundennamen füllen

Der gesamte VBA-Code der Sub-Prozedur *FillInvoiceForm* wird im Folgenden in 6 Teilen dargestellt:

```
Sub FillInvoiceForm(ByVal strShipNm As String)
' In Anlehnung an:
' http://msdn.microsoft.com/de-de/library/aa140287(v=office.10).aspx
' gefunden am 19.10.2014
' Verweis: Microsoft ActiveX Data Objects 6.1 Library
' Aufgabe: Einfaches Rechnungsformular mit Daten aus Northwind.mdb füllen
'         für einen von Benutzer ausgewählten Kundennamen.
' Lokale Variablen deklarieren
Dim rstInvoices As New Recordset ' Datensätze für Rechnungsdaten
' Word-Objektvariablen deklarieren
Dim objDoc As Word.Document ' Dokument
Dim objTable As Word.Table ' Tabelle
Dim objRow As Word.Row ' Zeile
Dim objCell As Word.Cell ' Zelle
Dim rngCell As Word.Range ' Zellbereich
' Sonstige Variablen deklarieren
Dim curSum As Currency ' Summe von €-Beträgen
Dim strCellContent As String ' Zellinhalt
Dim strUserTemplates As String ' Benutzervorlagen
```

Listing 8: Rechnungsformular füllen (1. Teil)

Listing 8 beinhaltet den Deklarationsteil der Prozedur *FillInvoiceForm*.

```
' Fehlerbehandlung vorsehen
On Error GoTo Err Point
' SQL-SELECT-Befehl vorgeben
p_strSQL = "SELECT * FROM Invoices WHERE Orders.ShipName='" & strShipNm & "'"
' Datensätze für Rechnungsdaten (aus Invoices) öffnen
rstInvoices.Open Source:=p_strSQL, ActiveConnection:=p_strConnect, _
CursorType:=adOpenStatic, LockType:=adLockReadOnly
If rstInvoices.RecordCount = 0 Then
MsgBox "Keine Rechnungsdaten gefunden", vbCritical, "Fill_Invoice_Form"
GoTo Exit_Point
End If
' Pfad der benutzerdefinierten Dokumentvorlagen festlegen
strUserTemplates = Application.Options.DefaultFilePath(wdUserTemplatesPath)
' Word-Dokument erstellen auf Basis der Benutzervorlage p_cstrTemplate
Set objDoc = Documents.Add(strUserTemplates & Application.PathSeparator & _
p_cstrTemplate)
If objDoc Is Nothing Then
MsgBox "Benutzervorlage nicht gefunden!", vbCritical, "Fill_Invoice_Form"
GoTo Exit_Point
End If
```

Listing 9: Rechnungsformular füllen (2. Teil)

Im Anschluss wird mit einem SQL-Befehl das *Recordset*-Objekt *rstInvoices* auf der Basis der *Invoices*-Abfrage in *Northwind.mdb* geöffnet, siehe Listing 9. Wenn dort keine Datensätze gefunden werden, endet die Prozedur. Der SQL-Befehl besitzt eine WHERE-Klausel, um die Datensätze für einen bestimmten Kunden abzurufen. Die Variable *strShipNm* enthält dessen Namen.

Anschließend wird ein Dokument auf Basis der Benutzervorlage *Simple\_Invoice.dotx* mithilfe der *wdUserTemplatesPath*-Enumeration der *DefaultFilePath*-Eigenschaft des *Options*-Objekts angehängt, siehe Listing 9. Wenn die Benutzervorlage nicht gefunden wird, endet die Prozedur.

## Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen

Dr. V. Thormählen

Als Nächstes wird mit dem Einsetzen der einzelnen Felder des *Recordset* in die Textmarken (engl. *bookmarks*) begonnen, die in der benutzerdefinierten Dokumentvorlage enthalten sind. Genau 11 Textmarken werden bedient, siehe Listing 10. Dies wird durch wiederholten Aufruf der *SetBookmark*-Funktion erledigt, die das *Range*-Objekt der *Bookmarks*-Auflistung verwendet:

```
' Daten für die Firmeninformation aus Northwind.mdb holen
' und in das Rechnungsformular einsetzen
SetBookmark ActiveDocument, "CoName", rstInvoices("Customers.CompanyName").Value
SetBookmark ActiveDocument, "CoAddress", rstInvoices("Address").Value
SetBookmark ActiveDocument, "CoCity", rstInvoices("City").Value
If IsNull(rstInvoices("Region").Value) Then
    SetBookmark ActiveDocument, "CoState", vbNullString
Else
    SetBookmark ActiveDocument, "CoState", rstInvoices("Region").Value
End If
SetBookmark ActiveDocument, "CoZip", rstInvoices("PostalCode").Value

' Daten für den Rechnungsempfänger aus Northwind.mdb holen
' und in das Rechnungsformular einsetzen
SetBookmark ActiveDocument, "BillToName", rstInvoices("Salesperson").Value
SetBookmark ActiveDocument, "BillToCompany", rstInvoices("ShipName").Value
SetBookmark ActiveDocument, "BillToAddress", rstInvoices("ShipAddress").Value
SetBookmark ActiveDocument, "BillToCity", rstInvoices("ShipCity").Value
If IsNull(rstInvoices("ShipRegion").Value) Then
    SetBookmark ActiveDocument, "BillToState", vbNullString
Else
    SetBookmark ActiveDocument, "BillToState", rstInvoices("ShipRegion").Value
End If
SetBookmark ActiveDocument, "BillToZip", rstInvoices("ShipPostalCode").Value
```

Listing 10: Rechnungsformular füllen (3. Teil)

Nach der Belegung der 11 Textmarken wird ein Verweis auf die letzte Tabelle in der Benutzervorlage erstellt. Die linke Spalte dieser Tabelle nimmt die jeweilige Produktbezeichnung auf und die rechte den dazugehörigen €-Betrag, siehe Listing 11.

Danach wird die Summe der €-Beträge in der 2. Spalte der aktuellen Tabelle bestimmt und in deren Fußzeile eingesetzt, siehe Listing Listing 12.

```
' Letzte Tabelle in Dokument (Rechnungspositionen) bestimmen
With ActiveDocument
  Set objTable = .Tables(.Tables.Count)
End With
' Format dieser Tabelle festlegen
With objTable
  .Style = "Tabellenraster"
  .ApplyStyleFirstColumn = False
  .ApplyStyleHeadingRows = True
  .ApplyStyleLastColumn = False
  .ApplyStyleLastRow = False
  With .Rows.First
    ' Kopfzeile wiederholen
    .HeadingFormat = True
    ' Kopfzeile grau hinterlegen
    With .Shading
      .Texture = wdTextureNone
      .ForegroundColor = wdColorAutomatic
      .BackgroundPatternColor = wdColorGray10
    End With
  End With
End With
' Kopfzeile der letzten Tabelle füllen mit Fettformatierung
p_intRow = 1
With objTable.Cell(p_intRow, 1).Range
  .Text = "Produkt"
  .Bold = True
End With
With objTable.Cell(p_intRow, 2).Range
  .Text = "Betrag"
  .Bold = True
End With
' Restliche Tabellenzeilen füllen mit Daten aus Northwind.mdb
Do While Not rstInvoices.EOF
  p_intRow = p_intRow + 1
  objTable.Cell(p_intRow, 1).Range.Text = rstInvoices("ProductName")
  objTable.Cell(p_intRow, 2).Range.Text = Format(rstInvoices("ExtendedPrice"),
"Currency")
  objTable.Rows.Add
  rstInvoices.MoveNext
Loop
' Zelleninhalte der 2. Spalte rechtsbündig ausrichten
For Each objRow In objTable.Rows
  objRow.Cells(2).Range.ParagraphFormat.Alignment = wdAlignParagraphRight
Next objRow
```

Listing 11: Rechnungsformular füllen (4. Teil)

```
' Summe der €-Beträge in der 2. Spalte der aktuellen Tabelle bestimmen
For Each objCell In objTable.Columns(2).Cells
  strCellContent = GetCellContent(objCell)
  ' Nur €-Beträge addieren
  If IsNumeric(strCellContent) Then
    curSum = curSum + CCur(strCellContent)
  End If
Next objCell
' Fußzeile der aktuellen Tabelle belegen mit der Summe der €-Beträge
With objTable.Rows.Last
  .Cells(1).Range.Text = "Spaltensumme"
  .Cells(1).Range.Bold = True
  .Cells(2).Range.Text = Format(curSum, "##,###.00 €")
  .Cells(2).Range.Bold = True
End With
```

Listing 12: Rechnungsformular füllen (5. Teil)

## Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen Dr. V. Thormählen

Am Ende der Prozedur wird eine Fertigmeldung ausgegeben und die WORD-Objektvariablen werden freigegeben, siehe Listing 13.

```
' Fertigmeldung ausgeben
MsgBox "Fertig!", vbExclamation, "Fill_Invoice_Form"

' Schreibmarke ins Kombinationsfeld setzen
ThisDocument.InlineShapes(1).Select
Exit_Point:
On Error Resume Next
' Word-Objektvariablen freigegeben
Set objDoc = Nothing
Set objTable = Nothing
Exit Sub
Err Point:
MsgBox "Fehler #: " & Err.Number & ", " & Err.Description, vbCritical,
"Fill_Invoice_Form"
Resume Exit_Point
End Sub
```

Listing 13: Rechnungsformular füllen (6. Teil)

### Modul mdlOtherProcs

In diesem Standardmodul befinden sich 6 Prozeduren. Aus Tabelle 3 ist ersichtlich, von welcher übergeordneten Prozedur sie jeweils aufgerufen werden.

Name der Prozedur	Typ der Prozedur	Aufrufende Prozedur(en)
SetBookmark	Sub	FillInvoiceForm
GetCellContent	Function	FillInvoiceForm
CheckInlineShapes	Function	Document_Open
CheckNumberOfBookmarks	Function	Document_Open
CheckNumberOfTables	Function	Document_Open
WordDateiSpeichernUndSchliessen	Sub	Document_Open cmdStopp_Click

Tabelle 3: Übersicht über aufgerufene und aufrufende Prozeduren

Die **gelb** hervorgehobene Kommentarzeile am Anfang einer Prozedur bezeichnet die jeweilige Aufgabe:

```
Sub SetBookmark(ByRef objDoc As Word.Document, ByVal strBookmark As String, ByVal
strValue As String)
' Textmarken belegen
If objDoc.Bookmarks.Exists(strBookmark) Then
objDoc.Bookmarks(strBookmark).Range.Text = strValue
Else
MsgBox "Textmarke nicht definiert: " & strBookmark, vbCritical,
"Fill_Invoice_Form"
End If
End Sub
```

```
Function GetCellContent(ByRef objCell As Word.Cell) As String
' Zellinhalt bestimmen
Dim objRng As Word.Range
Set objRng = objCell.Range
objRng.End = objRng.End - 1
GetCellContent = objRng.Text
Set objRng = Nothing
End Function
```

**Office Integration: Datensätze aus Access mit Word in ein einfaches Rechnungsformular einfügen**  
Dr. V. Thormählen

```
Function CheckInlineShapes() As Boolean
    ' Art der Steuerelemente im Dokument prüfen
    CheckInlineShapes = True
    With ThisDocument.InlineShapes
        If .Item(1).Type <> wdInlineShapeOLEControlObject Or _
            .Item(2).Type <> wdInlineShapeOLEControlObject Then
            CheckInlineShapes = False
            MsgBox "Zahl der Inline-Formen ist falsch!", vbExclamation
            Exit Function
        End If
    End With
End Function
```

```
Function CheckNumberOfBookmarks(ByVal intNbrBM As Integer) As Boolean
    ' Zahl der Textmarken in der benutzerdefinierten Dokumentvorlage prüfen
    Dim objTmp As Word.Document ' Dokumentvorlage
    Dim strUserTemplates As String ' Pfad zur Vorlage
    CheckNumberOfBookmarks = True
    strUserTemplates = Application.Options.DefaultFilePath(wdUserTemplatesPath)
    Set objTmp = Documents.Open(strUserTemplates & _
        Application.PathSeparator & p_cstrTemplate)
    If objTmp.Bookmarks.Count <> intNbrBM Then
        MsgBox "Zahl der Textmarken in der Dokumentvorlage '" & _
            p_cstrTemplate & "' ist falsch!", vbCritical, "CheckNumberOfBookmarks"
        CheckNumberOfBookmarks = False
    End If
    objTmp.Close SaveChanges:=wdDoNotSaveChanges
    Set objTmp = Nothing
End Function
```

```
Function CheckNumberOfTables(intNbrTbl As Integer) As Boolean
    ' Zahl der Tabellen in der benutzerdefinierten Dokumentvorlage prüfen
    Dim objTmp As Word.Document ' Dokumentvorlage
    Dim strUserTemplates As String ' Pfad zur Vorlage
    CheckNumberOfTables = True
    strUserTemplates = Application.Options.DefaultFilePath(wdUserTemplatesPath)
    Set objTmp = Documents.Open(strUserTemplates & _
        Application.PathSeparator & p_cstrTemplate)
    If objTmp.Tables.Count <> intNbrTbl Then
        MsgBox "Zahl der Tabellen in der Dokumentvorlage '" & _
            p_cstrTemplate & "' ist falsch!", vbCritical, "CheckNumberOfTables"
        CheckNumberOfTables = False
    End If
    objTmp.Close SaveChanges:=wdDoNotSaveChanges
    Set objTmp = Nothing
End Function
```

```
Sub WordDateiSpeichernUndSchliessen()
    ' Aktuelles Dokument speichern und schließen
    If Documents.Count > 0 Then
        With ActiveDocument
            If .Saved Then
                MsgBox "Die Word-Datei '" & .Name & "'" & vbCrLf & _
                    "ist bereits gespeichert.", vbInformation, _
                    "Word Datei speichern"
            Else
                .Save
            End If
            .Close
        End With
    Else
        MsgBox "Keine offene Word-Datei gefunden!", _
            vbExclamation, "Word Datei speichern"
    End If
End Sub
```

## Ergebnis

Mit den oben dokumentierten Prozeduren kann die Benutzervorlage (siehe Abbildung 1) für einen vorgegebenen Kundennamen gefüllt werden, siehe Abbildung 5. Das hier gezeigte einfache Rechnungsformular wurde beispielsweise erstellt für den Listeneintrag *Alfreds Futterkiste* des Kombinationsfelds *cboShipName*.

Das gezeigte einfache Rechnungsformular kann beliebig verfeinert werden, um die Praxistauglichkeit zu verbessern.

**Einfaches Rechnungsformular**

**Kunde**

Alfreds Futterkiste
Obere Str. 57
Berlin
12209

**Rechnungsempfänger**

Margaret Peacock
Alfred's Futterkiste
Obere Str. 57
Berlin
12209

**Rechnung**

Produkt	Betrag
Aniseed Syrup	60,00 €
Lakkalikööri	270,00 €
Raclette Courdavault	825,00 €
Original Frankfurter grüne Soße	20,80 €
Grandma's Boysenberry Spread	380,00 €
Rössle Sauerkraut	91,20 €
Escargots de Bourgogne	503,50 €
Fløtemysost	430,00 €
Aniseed Syrup	60,00 €
Lakkalikööri	270,00 €
<b>Spaltensumme</b>	<b>2.910,50 €</b>

Abbildung 5: Automatisch ausgefülltes Rechnungsformular für Alfreds Futterkiste

Der Projekt-Explorer enthält nach dem Füllen des einfachen Rechnungsformulars folgende Ergänzungen.

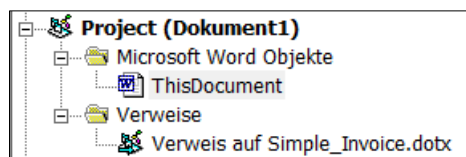


Abbildung 6: Dokument1 vor dem Speichern im Projekt-Explorer

## **Ergänzungen**

Die am Anfang des vorliegenden Dokuments enthaltenen ActiveX-Steuerelemente (siehe erste Seite unterhalb der Kopfzeile) sind nach einem Bildlauf nicht mehr sichtbar. Dieser Nachteil lässt sich vermeiden, wenn diese Steuerelemente in einem sog. *Benutzerformular* (engl. *UserForm*) angezeigt werden. Dieses lässt sich dann – wie ein gewöhnliches WORD-Dialogfeld – auf dem Bildschirm einblenden.