

## Mit einem Grid viel Zeit sparen

Artikel	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Gebackte	Linsen	Lebkuchen	10,00 €	2	20%	8,00 €
Blanc	30	Schokolade	10,00 €	21	25%	14,75 €
19945	30	Chokolade ohne	10,00 €	21	25%	14,75 €
19990	30	Rosine-Schokolade	10,00 €	15	25%	15,00 €

Exotic Liquids
Exotic Liquids
Forêts d'arbres
Formaggi Fortini s.r.l.
Gai pâturage
G'dey, Mele
Grandma Kelly's Homestead
Heli Süßwaren GmbH & Co. KG
Karkki Oy



Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich

Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich
Artikelname	10645	Bezeichnung	12.Aug.1997	Lieferzeit	22.Sep.1997	Vergleich

Artikel	Einzelpreis	Anzahl	Robott	Endpreis
Spiegelbild	6,00 €	2	25%	9,00 €
Chokolade ohne	9,00 €	21	25%	14,75 €
Rosine-Schokolade	22,00 €	15	25%	25,50 €



## Mit einem Grid viel Zeit sparen

### Robert Bollig (44), Frankfurt am Main

Mr. „On Error Resume Next“



Seit 1992 bin ich in der EDV tätig und mit Access 97 begann ich die professionelle Datenbankerstellung sowie die Entwicklung eigener Programme. Noch immer bin ich der 97er Runtime treu, denn sie läuft und läuft und läuft . . . auf allen Windows-Versionen - Letztes Update 2006.



*Ich liebe Lösungen, die sich immer wieder Erweitern lassen indem Kunden neue Wünsche einfach einpflegen...*

#### Frontends

Microsoft Access/ Office  
WEB, Wordpress und andere CMS-Systeme, VB.Net, C++, C#

#### Backends

Microsoft SQL-Server, Oracle, mySQL und Access-BE im Netzwerk (~1 Mio Daten/ 1,8 GB)

## Erwischt!

Sie lesen also auch schon vor dem Vortrag die Unterlagen?!

Cool!

Denn ich werde den Vortrag genau da ansetzen wo diese Seiten aufhören.



Und Sie kennen dann schon die grundsätzlichen Befehle, die wir in den Demos immer wieder verwenden und mit denen wir sooo viel Zeit sparen werden.

Die auf den kommenden Seiten angerissenen Punkte, sind dabei nicht wirklich richtige Probleme, kennen wir ja alle schon entsprechende Workarounds. Vielmehr möchte ich Ihnen damit eine kleine Hilfestellung mitgeben, wann der Einsatz eines iGrids Sinn machen könnte.

Im Anschluss daran habe ich ein paar kleine Code-Schnipsel zusammengetragen, mit denen schon eine Vielzahl von Anforderungen umgesetzt werden kann.

Fallen Ihnen weitere Einsatzszenarien für ein Grid ein, so geben Sie doch bitte Bescheid, ich nehme diese dann - wie auch andere Ideen/ Workarounds - in die AEK-Downloads auf.

## Listboxen

Im Grunde haben sich diese seit Jahren ja nicht verändert. Immer wieder braucht es einige Tricks, bzw. in einer mde/accd sind manche Dinge gar nicht machbar.

- Texte ausrichten/ formatieren
- Bedingte Formatierung/en
- Spalten-Titel Ein-/Ausblenden
- Wechsel zw. Einfach-/ Mehrfachauswahl
- Reihenfolge der Spalten ändern
- Fixierung von Spalten (z.B. wie in Excel)
- Sortieren und Gruppieren
- Zwischensummen oder Zählungen

## Unterformulare

Sind UFOs erste Wahl wird es immer wieder spannend, wenn neue Felder dazukommen, einige Wegfallen oder ein (neues) Rechte-System integriert werden will. Der Aufwand ist dann nicht unerheblich und einige Punkte können dann mitunter mehr Arbeit erzeugen als gedacht.

- Änderung von Beschriftungen  
(kommt ja auch mal ohne Mehrsprachigkeit vor)
- Sperren oder Freigeben von Controls
- Änderungen von Funktionen
- Erweiterung einer Suchfunktion
- Gruppierung oder Zwischensummen

## Treeview

Inzwischen gibt es so viele Ideen und Alternativen, dass man schnell den Überblick verlieren kann. Einige Gemeinsamkeiten haben jedoch quasi alle.

- Es werden immer weitere Controls/ UFOs zur Anzeige der Daten benötigt
- Ladezeiten und Programmieraufwand sind manchmal grausam
- Es braucht neuen/ eigenen Code zum Laden oder Abfangen von Ereignissen.
- ...

## Combobox

Haben sich diese je wirklich verändert?

- Keine Volltextsuche (Autovervollst. nur am Wortanfang)
- Keine Gruppierungen (optisch)
- Abh. Comboboxen im Endlosformular
- ...

## Menü / Ribbon

Wer auf Menüs oder Ribbons verzichten mag / will / muss, muss sich nach diversen Alternativen umschauen und landet oft bei Menu-, o. Navigationbars.

- Oft weitere Controls erforderlich
- Code kann ggf. nicht genutzt werden
- Implementierung CI
- Berechtigungskonzepte integrieren
- Einbau einer Mehrsprachigkeit
- ...

## Einfache Dateneingabe

- Kleine Formulare, die nur mit ein paar Controls bestückt sind
- Beschriftung, Inhalt und Code muss immer wieder angepasst werden
- PopUp-Auswahlformulare
- Kalender mit Zusatzinfos
- ...

## iGrid-Basics

Das iGrid enthält bereits ein Standard-Kontextmenü, welches beim Klicken auf Spaltenköpfe Optionen zum Gruppieren und Sortieren anbietet. Beim Klicken auf eine Gruppierung oder Zelle/ Zeile, bietet es kontextsensitive Befehle zur Datenbearbeitung.

Die Voreinstellung aller Texte & Meldungen ist Englisch. Sie lassen sich via Code ändern.

Kontextmenüs können jederzeit neu erstellt oder geändert/ erweitert werden. Sie können so auf Bereich und Inhalt entsprechend reagieren.

With iGrid.ContextMenuCustomItems ( **igContextMenuCell** )

.Add „Dies ist ein Menüpunkt, beim Klick auf eine **Zelle/ Zeile**“

.Add „Und hier kommt noch einer...“

End With

With iGrid.ContextMenuCustomItems ( **igContextMenuColHeader** )

.Add „Dies ist ein Menüpunkt, beim Klick auf einen **Spaltenkopf**“

End With

## iGrid-Basics

Die folgenden Beispiele dienen im Grunde eher der Veranschaulichung und der schnellen Einarbeitung, damit wir gemeinsam eine Datenbank während des Vortrags erstellen können.

**Mit den kleinen Code-Schnipseln können wir das iGrid jedoch schon einsetzen.**

Die **hervorragende Hilfe-Datei** - ja es ist eine **CHM-Datei** ☺ - beinhaltet viele Beispiele, die man oft 1:1 übernehmen kann. Inhaltsverzeichnis & Suche sind sinnvoll aufgebaut.

Das iGrid ist eine Matrix. Geben Sie die gewünschte Anzahl Spalten & Zeilen vor, können Sie schon loslegen. Mit einem Schleifen-Konstrukt Ihrer Wahl können Sie also Daten ins iGrid Schreiben. Zahlen & Texte können Sie dabei innerhalb einer Spalte jederzeit mischen.

Mit **FillFromRecordset** & **LoadFromArray** können wir - mit quasi einer Zeile - das iGrid füllen.

Mit **AddCol** & **AddRow** können wir sehr detailliert Zeilen & Spalten erstellen und einfügen.

## iGrid-Basics

Inhalte der Zellen Schreiben & Lesen Sie mit `.CellValue ( Zeile, Spalte )`, diese adressieren Sie mit ihrer Nummer oder dem sogenannten **ColKey**.

With iGrid

```
.ColCount = 2
.RowCount = 3
```

```
.CellValue(1, 1) = "abc"
.CellValue(2, 1) = 123
```

End With

Col 1	Col 2	
abc		
123		

Spalten & Zeilen sind dabei 1-basierend, was man beim Durchlaufen Null-basierter Arrays oder (Recordset-)Loops beachten muss.

Da wir auch mit den KEYS die Koordinaten einer Zelle angeben können, sind Konstrukte wie `.CellValue( i, „Lagerbestand“)` möglich. Code-Anpassungen sind dann nicht mehr erforderlich, wenn sich die (originäre) Spaltenreihenfolge beim Laden der Daten ändert.

## iGrid-Basics

```
Sub FillFromRS ( ByVal RS As Object, _
    Optional ByVal eMode As EFillFromRSMODE = 1,
    Optional ByVal vRowKeyField As Variant)
```

1 = igFillRSColsIfEmptyAndRows

```
Set daoRS = FE.OpenRecordset("select * from Artikel")
```

With iGrid

```
.FillFromRS daoRS, 1
```

```
.GroupObject.Clear
.GroupObject.AddItem "Auslaufartikel", 1
.Group
```

```
.CollapseAllChildRows (1)
```

End With

Artikel-Nr	Artikeln...	Lieferan...	Kategor...	Lieferei...
+ Auslaufartikel: Falsch				
- Auslaufartikel: Wahr				
42	Singap...	20		5 32 x 1-k...
53	Perth P...	24		6 48 Stück
5	Chef Ant...	2		2 36 Karto...
29	Thüring...	12		6 50 Beut...
17	Alice M...	7		6 20 x 1-k...
...	...	...		...

## iGrid-Basics

With iGrid

```
.GroupObject.Clear
.GroupObject.AddItem „1. Gruppierung LAND“, igSortAsc
.Group
```

End With

Gruppierungen können mit wenigen Zeilen erstellt werden, die Properties sind selbsterklärend:

**AllowGrouping, PrefixGroupValue, DefaultAutoGroupRowExpanded, RowIsGroup(vRow), ...**

Mit Events können wir uns nahezu jederzeit einklinken:

```
BeforeContentsGrouped(), AfterContentsGrouped()
AfterAutoGroupRowCreated( ByVal IRow As Long, ByVal IItemCnt As Long, ...
BeforeRowCollapseExpand( ByVal IRowIfAny As Long, ByVal bNowExpanded As Boolean, ...
```

Fernsteuern (von Außen) geht auch:

```
.CollapseAllChildRows (1), .ExpandAllChildRows(21)
```

## iGrid-Basics

Alle Grundeinstellungen des iGrids lassen sich jederzeit ändern; alle Einstellungen sind beliebig kombinierbar.

With iGrid

```
.RowMode = True
```

```
.MultiSelect = True
```

End With

Auslaufartikel: Wahr			
42 Singep...	20	5	32 x 1-k...
53 Perth P...	24	6	48 Stück
5 Chef Ant...	2	2	36 Karto...
29 Thüring...	12	6	50 Beut...
17 Alice M...	7	6	20 x 1-k...

Auslaufartikel: Wahr			
42 Singep...	20	5	32 x 1-k...
53 Perth P...	24	6	48 Stück
5 Chef Ant...	2	2	36 Karto...
29 Thüring...	12	6	50 Beut...
17 Alice M...	7	6	20 x 1-k...

Auslaufartikel: Wahr			
42 Singeporen Ho...	20	5	32 x 1-k...
53 Perth Pastes	24	6	48 Stück
5 Chef Anton's Gu...	2	2	36 Karto...
29 Thüringer Rostbr...	12	6	50 Beut...
17 Alice Mutton	7	6	20 x 1-k...
9 Misch Kebab N...	4	6	18 x 500...

Sie können so aber auch ein Grid mehrfach verwenden, in dem Sie beim Öffnen angeben, ob eine Mehrfachauswahl zulässig ist, ob Zeilen komplett markiert werden sollen, usw.

## iGrid-Basics

```
Sub LoadFromArray( ByVal vStartRow As Variant, _
                  ByVal vStartCol As Variant, _
                  ByRef vArray As Variant, _
                  Optional ByVal bColMajorOrder As Boolean = False)
```

‘Lade ein 2-Dim. Array Ab Zeile 1 in der 1. Spalte beginnend  
 Dim arr(1 To 10, 1 To 5) As Variant  
 ...  
 iGrid.LoadFromArray 1, 1, arr

‘Lade ein 1-Dim. Array: Ab Zeile 1 in die 3. Spalte  
 Dim arr(1 To 15) As Long  
 ...  
 iGrid.LoadFromArray 3, 1, arr

## iGrid-Basics

```
Sub Function AddCol( _
  Optional ByVal sKey As String, _
  Optional ByVal sHeader As String, _
  Optional ByVal lWidth As Long = -1, _
  Optional ByVal eHeaderAlignH As EHAlignment..., _
  Optional ByVal eHeaderAlignV As EVAlignment ..., _
  Optional ByVal eHeaderTextFlags As ETextForma..., _
  Optional ByVal iHeaderIcon As Integer = -1, _
  Optional ByVal bVisible As Boolean = True, _
  Optional ByVal vColBefore As Variant, _
  Optional ByVal eSortType As ESortTypes = igSortBy..., _
  Optional ByVal eSortOrderDefault As ESortOrders..., _
  Optional ByVal lMinWidth As Integer = -1, _
  Optional ByVal lMaxWidth As Integer = -1, _
  Optional ByVal bAllowSizing As Boolean = True, _
  Optional ByVal vTag As Variant, _
  Optional ByVal lCount As Long = 1 _
) As CellObject
```

```
Sub AddRow( _
  Optional ByVal sKey As String, _
  Optional ByVal vRowBefore As Variant, _
  Optional ByVal bVisible As Boolean = True, _
  Optional ByVal lHeight As Long = -1, _
  Optional ByVal bGroupRow As Boolean = False, _
  Optional ByVal vRowParent As Variant, _
  Optional ByVal bExpanded As Boolean = True, _
  Optional ByVal btLevel As Byte = 0, _
  Optional ByVal eTreeButton As ERowTreeButton ..., _
  Optional ByVal bVisibleAsChild As Boolean = True, _
  Optional ByVal vTag As Variant, _
  Optional ByVal lCount As Long = 1 _
)
```

Die Beiden lassen wir mal so stehen ☺

## iGrid-Basics

Fast vergessen: Die TreeView-Funktionalität

Eigentlich ist das iGrid mehr ein TreeGrid als ein TreeView, können Sie auch bei Nutzung der TreeCol-Funktionalität Ihre Daten(-Zeilen) anzeigen und direkt bearbeiten lassen.

Methoden und Events sind quasi gleich. Nur das Laden der Daten müssen Sie ein klein wenig anders gestalten. Und wenn Sie nur eine Spalte (die TreeCol) anzeigen, erhalten Sie ein „einfaches TreeView“. Für das Erstellen einer TreeCol-Spalte gibt es mehrere Wege.

With iGrid

```
.AddCol IWidth:=120 : .AddCol:
.TreeCol = 1
.AddRow: .CellValue(.RowCount, 1) = "Node-1"
.AddRow vRowParent:=1: .CellValue(.RowCount, 1) = "Node-1.1.,
.AddRow vRowParent:=1: .CellValue(.RowCount, 1) = "Node-1.2"
```

End With

Node-1	
Node-1.1	
Node-1.2	

## iGrid-Basics

Beide Beispiele erzeugen dieselbe Struktur. Beispiel 2 zeigt jedoch auf, was im Grunde im Hintergrund passiert: Es werden die entsprechenden RowLevel und Sichtbarkeiten gesetzt.

With iGrid

```
.AddCol IWidth:=120: .AddCol: .AddCol
.TreeCol = 1
.RowCount = 3

.CellValue(„Zeile 1“, 1) = "Node-1,,: .RowTreeButton(1) = igRowTreeButtonVisible
.CellValue(„Zeile 2“, 1) = "Node-1.1,,: .RowTreeButton(2) = igRowTreeButtonHidden
.RowLevel(„Zeile 2“) = 1:
.CellValue(„Zeile 3“, 1) = "Node-1.2,,: .RowTreeButton(3) = igRowTreeButtonHidden
.RowLevel(„Zeile 3“) = 1:
```

End With

Sie können also auch erst die Daten laden und dann anschließend die Struktur erzeugen.