

Tipp: Das automatische Drehfeld in Microsoft Excel

von Ignatz Schels

Trotz allem Komfort, den Microsoft Excel bietet, fehlt ein nützliches Feature: Ein Drehfeld am Zellrand, mit dem sich der Wert in der Zelle per Mausklick erhöhen oder verringern lässt (Bild 1). Das spart viel Zeit bei der Aktualisierung von Excel-Blättern mit großen Zahlenmengen, da sich die Werte alleine mit der Maus ändern lassen, ohne die Tastatur benutzen zu müssen.

Mit zwei kleinen Makros lässt sich diese Aktion einfach umsetzen, Programmierkenntnisse sind dafür nicht erforderlich. Für das hier vorgestellte Beispiel reicht es aus, wenn Sie wie unten beschrieben den gelisteten Makrocode (siehe auch Datei "Makrocode.txt") an die entsprechende Stelle in der Arbeitsmappe kopieren.



Ignatz Schels

selbstst. DV-Dozent und Journalist, konzipiert und leitet MS Project-Seminare, Projektberater und Autor

Kontakt: info@schels.de

Mehr Informationen unter: www.projektmagazin.de/autoren/

	A	B	C	D
1	Arbeitspaket	Soll-Kosten (in TEU)	Ist-Kosten (in TEUR)	Abweichung (in TEUR)
2	Aushub, Tiefbau	35	34	1,0
3	Bodenfläche, Unterbau	12	15	3,0
4	Keller	21	25	4,0
5	Estricharbeiten	8	6	2,0
6	Erdgeschoß mit Eingang	6	5	1,0
7	Dachgeschoß	10	10	0,0
8	Sanitär/Elektroinstallation	11	10	1,0
9	Heizung	12	15	3,0
10	Malerarbeiten	9	8	1,0

Bild 1: Das Drehfeld erscheint automatisch, sobald eine Zelle mit Zahlenwert markiert wird.

Die einzige Einschränkung ist, dass Drehfelder nur ganze Zahlen im Zahlenbereich von 0 bis 30.000 schreiben können. Die Schrittweite, mit der sie den Inhalt der verknüpften Zelle erhöhen oder verringern, lässt sich in den Eigenschaften festlegen. (Mehr Informationen zu Formularelementen und Drehfeldern: Aus der Excel-Trickkiste. Fortgeschrittene Techniken für Projektleiter. Teil 1: Ein Projektplan mit Teamauswahl, Projekt Magazin 20/2010 und Teil 2: Portfolio-Auswahl mit dynamischen Diagrammen, Projekt Magazin 21/2010)

Funktionsweise des Makros

Die Funktionsweise des Makros ist verhältnismäßig einfach: "SelectionChange" heißt das Ereignis, das mit der Positionierung des Zellzeigers ausgelöst wird. Es wird so programmiert, dass jede markierte Zelle automatisch ihr Drehfeld am rechten Rand erhält. Der Trick dabei: Als Position des Drehfelds wird einfach die linke obere Ecke der nächsten Zelle verwendet. "Target" ist der Objektname der Zelle, "target.next" liefert die Informationen über die nächste Zelle.

Aber Vorsicht: Da das Drehfeld nur dann erscheinen soll, wenn die markierte Zelle einen Zahlenwert enthält, muss das Makro zuerst prüfen, ob der Zelleninhalt numerisch ist. Außerdem darf der Zellzeiger nicht in der letzten Spalte des Tabellen-Blatts stehen, da das Drehfeld dann nicht in der Ecke des nächsten Felds angezeigt werden kann. Auch bei verbundenen Zellen oder einer Mehrfachauswahl gibt es Probleme. Und natürlich darf die Zelle keine Formel enthalten, denn das Drehfeld würde diese überschreiben. Alle diese Fälle prüft das Makro ab. Tritt einer der genannten Fälle auf, springt es ohne das Drehfeld anzuzeigen an das Ende ("no Show").

Listing 1 zeigt den Makrocode mit Kommentaren zu den einzelnen Anweisungen.

Die Anweisung "Option Explicit" in der ersten Zeile stellt sicher, dass alle benutzten Variablen mit DIM deklariert sind. Excel fügt Sie automatisch in neue Module ein, wenn unter *Extras / Optionen* die Option "Variablen Deklaration erforderlich" gesetzt ist.

! Ereignismakros, also Makros, die nicht bewusst durch den Anwender, sondern durch ein Ereignis angestoßen werden (in unserem Fall durch das Markieren einer Zelle), greifen in Prozesse wie "Kopieren" und "Einfügen" ein. Sie können kritisch sein, wenn die darin programmierten Anweisungen Daten eintragen oder das Tabellenlayout ändern. Setzen Sie im Makro aus diesem Grund immer einen Schalter, der die Aktionen bei Bedarf abschaltet. Ein Eintrag in eine freie Zelle fernab des Auswertungsbereichs genügt für diesen Zweck, wenn das Makro eine Anweisung enthält, die ermittelt, ob dieser Eintrag gesetzt ist.

Im Beispiel lautet die zugehörige Anweisung:

```
If Range("IV1").Value = "no spB" Then
GoTo noShow
```

Sie steht am Anfang des Makros und prüft, ob die Zelle \$IV\$1 den Eintrag "no spB" enthält. Ist das der Fall, beendet die Anweisung das Makro.

Makrocode einfügen

Und so fügen Sie den Code in das gewünschte Tabellenblatt ein:

Aktivieren Sie in Excel zunächst mit Alt+F11 den Visual Basic-Editor. Dort sehen Sie im linken Bereich den Projekt-Explorer, in dem die Arbeitsmappe als Projekt steht ("Mappe1" in Bild 2). Klicken Sie im zugehörigen Ordner "Microsoft Excel Objekte" doppelt auf den Namen des Tabellenblatts, für das das Makro gelten soll. Im Hauptfenster des VBA-Editors erscheint daraufhin das (zunächst leere) Modul für die Ereignismakros dieses Tabellenblatts. Kopieren Sie den Makrocode in das leere Fenster und beenden Sie den VBA-Editor anschließend mit Alt+Q (alternativ mit dem Befehl *Datei/Schließen und zurück zu Microsoft Excel*).

Das Makro – und damit das automatisch erzeugte Drehfeld – steht Ihnen jetzt im entsprechenden Tabellenblatt zur Verfügung. Um es bei Bedarf zu deaktivieren, tragen Sie "no spB" in Zelle "IV1" ein.

Option Explicit

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
Dim rngCell, spB As Shape, spbBreite As Integer
' Schalter
If Range("IV1").Value = "no spB" Then GoTo noShow
' Wenn mehr als eine Zelle markiert ist,
' bei verbundenen Zellen oder wenn die Zelle Text enthält
' oder wenn der Wert größer als 30.000 ist
If Target.Cells.Count > 1 _
Or Not IsNumeric(Target.Value) _
Or Target.Column = Target.End(xlToRight).Column _
Or Target.HasFormula _
Or Target.Value > 30000 Then GoTo noShow
' Breite des Drehfeldes festlegen
spbBreite = 15
' Aufruf der Funktion zum Löschen des Drehfelds
delShape
rngCell = Target.Address
' Drehfeld anlegen, Name zuweisen und Adresse aus der Zelle holen
Set spB = ActiveSheet.Shapes.AddFormControl _
(xlSpinner, Target.Next.Left, Target.Next.Top, spbBreite, Target.Next.Height)
With spB
.Name = "spbDrehfeld"
.ControlFormat.Value = Target.Value
.ControlFormat.LinkedCell = Target.Address
End With
Exit Sub
noShow:
delShape
End Sub

Function delShape()
' Drehfeld wird gelöscht, falls vorhanden
On Error Resume Next
ActiveSheet.Shapes("spbDrehfeld").Delete
On Error GoTo 0
Application.ScreenUpdating = True
End Function
```

Listing 1: Makrocode, um ein automatisches Drehfeld zu erzeugen.

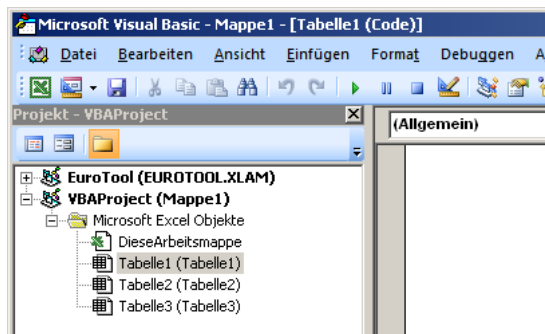


Bild 2: Die Arbeitsmappe als VBA-Projekt im Visual Basic Editor.

Hat Ihnen dieser Artikel gefallen?

Bewerten Sie ihn im Projekt Magazin online und teilen Sie so Ihre Meinung anderen Lesern mit. Wählen Sie dazu den Artikel im Internet unter <http://www.projektmagazin.de/archiv> oder klicken Sie [hier](#), um direkt zum Artikel zu gelangen.