
HARALD NAHRSTEDT

Excel + VBA

Ergänzungen

Kapitel

Einführung in VBA

Filter in VBA nutzen

Erstellt am 15.10.2011

Beschreibung Der Autofilter auf einer Excel-Tabelle bietet eine einfache Möglichkeit, Daten nach einem oder mehreren Spaltenkriterien einzuzugrenzen. Der Autofilter kann auch über VBA gesteuert werden.

Der Autofilter auf einer Excel-Tabelle bietet eine einfache Möglichkeit, Daten nach einem oder mehreren Spaltenkriterien einzugrenzen. Der Autofilter kann auch über VBA gesteuert werden. Bezogen auf die Objekte, gibt es dazu unterschiedliche Methoden.

Objekt.Methode	bewirkt/ergibt	Hinweis
<i>Range</i> .Autofilter	Ist kein Autofilter eingerichtet, dann wird der Autofilter installiert, sonst wird der vorhandene Autofilter gelöscht.	Nur ein Autofilter je Tabellenblatt möglich.
<i>Worksheet</i> .AutoFilterMode	Gibt "True" zurück, wenn Autofilter eingerichtet.	
<i>Worksheet</i> .FilterMode	Gibt "True" zurück, wenn im Blatt ein Autofilter eingerichtet und aktiv ist. Gibt "False" zurück, wenn im Tabellenblatt kein AutoFilter eingerichtet ist oder wenn der AutoFilter nicht aktiv ist.	
<i>Worksheet</i> .ShowAllData	Blendet alle Zeilen wieder ein.	Fehlermeldung, wenn Filterung nicht aktiv.

1 Die Methode Autofilter

Der Aufruf des Autofilters funktioniert wie ein Schalter. Ist der Autofilter aktiv, wird er ausgeschaltet, sonst eingeschaltet. Vor dem Aufruf wird eine beliebige Zelle in der Tabelle markiert. Bei dem Aufruf werden dann alle genutzten umliegenden Zellen gesucht und so die Tabelle gefunden. Der Autofilter wird danach in der obersten Zeile des markierten Bereichs installiert.

Name	Vorname	Geburtsdatum	Reisekosten
Müller	Maria	15.03.1982	221,45
Schmid	Anna	03.07.1974	1.456,33
Becker	Heike	08.03.1991	2.555,10
Schuster	Sonja	05.12.1981	877,45
Bauer	Zoe	05.06.1980	1.236,55
Schweitzer	Hans	07.07.1977	1.110,98
Bötcher	Lukas	05.05.1984	556,00
Brenner	Walter	06.09.1995	3.145,76
Schneider	Bernd	27.09.1990	228,88
Kutscher	Heiko	03.08.1985	112,45

Das Diagramm zeigt eine Excel-Tabelle mit vier Spalten: Name, Vorname, Geburtsdatum und Reisekosten. Die Spaltenüberschriften sind durch Dropdown-Pfeile markiert. Ein gelber Kasten mit der Aufschrift 'Gefundene Kopfzeile' zeigt auf die Spaltenüberschriften. Ein roter Pfeil zeigt auf die Zelle '05.06.1980' in der Spalte 'Geburtsdatum', die als 'markierte Zelle vor dem Autofilter-Aufruf' beschriftet ist.

Die nachfolgende Prozedur schaltet den Autofilter ein/aus.

```

Sub AutofilterToggle()
  ' Autofilter gibt's nur in Tabellenblättern:
  If TypeName(ActiveSheet) <> "Worksheet" Then
    Beep
    Exit Sub

  ' Falls kein Autofilter gesetzt ist, muss die aktive Zelle
  ' einen Wert haben, damit ein Autofilter gesetzt werden kann:
  ElseIf ActiveSheet.AutoFilterMode = False And _
    IsEmpty(ActiveCell) Then
    MsgBox "Autofilter kann nicht in einer leeren Zelle " & _
      "eingerrichtet werden !", vbCritical, "Fehler:"
    Exit Sub
  End If
  ' Falls Autofilter gesetzt, Filter löschen; falls nicht
  ' gesetzt, Filter einrichten:
  ActiveCell.AutoFilter
End Sub

```

Der Versuch, einen Autofilter auf einem anderen Objekt als auf einem Worksheet einzurichten, führt ebenso zu einer Fehlermeldung wie der Versuch, den Autofilter aufzurufen, während eine leere Tabellenzelle markiert ist.

2 Die Methode AutoFilterMode

Die Methode liefert den Wert „True“, wenn die Filter-Pfeile angezeigt werden. Dann lassen sich die Pfeile mit der Zuordnung von „False“ entfernen. Umgekehrt geht das jedoch nicht. Dazu muss die Methode *AutoFilter* verwendet werden.

3 Die Methode FilterMode

Mit dieser Methode kann nur ein Status abgefragt werden. Die Methode liefert dann den Wert „True“, wenn ein Autofilter auf dem Tabellenblatt existiert und auch gesetzt ist. Diese Methode dient nur diesem Zweck und damit können dann wieder alle Zeilen angezeigt werden.

```

Sub AlleDatenZeigen()
  If TypeName(ActiveSheet) <> "Worksheet" Then
    MsgBox "Falsches Objekt gewählt!", vbOKOnly & vbCritical
  ElseIf ActiveSheet.FilterMode = False Then
    MsgBox "Filter nicht angewendet!", vbOKOnly & vbCritical
  Else
    ActiveSheet.ShowAllData
  End If
End Sub

```

4 Filterdaten lesen

Die nachfolgende Prozedur liefert alle Daten der gesetzten Filter.

```

Sub NotiereAutoFilter()
  Dim filTab1 As Filter
  Dim wshTab1 As Worksheet

```

```
Dim wshTab2 As Worksheet
Dim iRow As Integer
Dim sKrit1 As String
Dim sKrit2 As String
Dim sOp As String

Const ns As String = "Not set"

Set wshTab1 = Worksheets("Tabelle1")
Set wshTab2 = Worksheets("Tabelle2")
wshTab2.Cells.ClearContents
On Error Resume Next
iRow = 1
For Each filTab1 In wshTab1.AutoFilter.Filters
    If filTab1.On Then
        sKrit1 = _
            Right(filTab1.Criteria1, Len(filTab1.Criteria1) - 1)
        If filTab1.Operator Then
            sOp = filTab1.Operator
            If sOp < 3 Then
                sKrit2 = _
                    Right(filTab1.Criteria2, Len(filTab1.Criteria2) - 1)
            Else
                sKrit2 = "-"
            End If
        Else
            sOp = ns
            sKrit2 = ns
        End If
    Else
        sKrit1 = ns
        sOp = ns
        sKrit2 = ns
    End If
    wshTab2.Cells(iRow, 1) = sKrit1
    wshTab2.Cells(iRow, 2) = sOp
    wshTab2.Cells(iRow, 3) = sKrit2
    iRow = iRow + 1
Next
wshTab2.Activate
Set wshTab2 = Nothing
Set wshTab1 = Nothing
End Sub
```

Angewandt auf unser Beispiel in Tabelle 1

Name	Vorname	Alter
Schmid	Anna	45
Becker	Heike	32
Bauer	Zoe	38
Schweitzer	Hans	41
Bötcher	Lukas	33

zeigt Tabelle2 die gesetzten Attribute und Operatoren. Die Spalte 1 zeigt als erstes Kriterium, dass nur alle Namen mit dem Anfangsbuchstaben B (B*) angezeigt werden. Die Spalte 2 zeigt mit einem Wert 2, dass der Operator Oder gesetzt ist und ein zweites Kriterium alle Namen mit dem Anfangsbuchstaben S (S*) zulässt.

	A	B	C
1	B*		2 S*
2	Not set	Not set	Not set
3	30	Not set	Not set

Die zweite Spalte in der Tabelle (Vorname) besitzt kein Kriterium. Die dritte Spalte Alter zeigt als einziges Kriterium alle Personen die älter als 30 sind.

5 Daten filtern

Die Syntax für die Autofilter-Methode lautet:

```
Range.AutoFilter(field, criteria1, operator, criteria2)
```

Darin ist

- **field** die fortlaufende Nummer der zu filternden Spalte im Autofilter-Bereich.
- **criteria1** ein Filterbegriff, z.B. "=B*"
- **operator** "xlAnd" oder "xlOr" für Verknüpfung mit "criteria2"; "xlTop10Items" gibt jene Anzahl an Datenzeilen zurück, die in "criteria1" mit einer Zahl angegeben werden. Zum Beispiel: "9":

```
Sub Makro2()
    Selection.AutoFilter Field:=1, Criteria1:="9", Operator:= _
        xlTop10Items
End Sub
```

- **criteria2** zweiter Filterbegriff, z.B. "=S*"

Alle Operatoren können auf Texte, Zahlen und Datumswerte angewendet werden. Die Umlaute werden bei deutscher Ländereinstellung von Windows korrekt behandelt. Groß- und Kleinschreibung sind gleichgesetzt.

Die nachfolgende Tabelle zeigt noch einmal alle Vergleichsoperatoren:

Vergleichs-Operator	Angezeigt werden die Datenzeilen,	Beispiel
=	in denen in der Spalte „field“ der Filterbegriff komplett enthalten ist	=Müller =123,12 =01.04.96
>	deren Wert in der Spalte "field" größer ist als im Filterbegriff angegeben.	>123,12 >Hofer
<	Es werden nur jene Datenzeilen gezeigt, deren Wert in der Spalte "field" kleiner ist als im Filterbegriff angegeben.	<01.04.1996 <123,12 <Österreich
>=	Es werden nur jene Datenzeilen gezeigt, deren Wert in der Spalte "field" größer ist als im Filterbegriff angegeben.	>=01.01.96 >=123,12 >=Österreich
<=	Es werden nur jene Datenzeilen gezeigt, deren Wert in der Spalte "field" kleiner ist als im Filterbegriff angegeben.	<=01.01.96 <=123,12 <=Österreich
<>	Es werden nur jene Datenzeilen gezeigt, deren Wert in der Spalte "field" nicht dem im Filterbegriff angegebenen Wert entspricht.	<>01.01.96 <>123,12 <>Österreich

Zu beachten ist, dass zusätzlich zu den Filterbegriffen auch die Vergleichs-Operatoren in der Zeichenkette mit dem Filterbegriff an die Autofilter-Methode übergeben werden muss. Das nachfolgende Beispiel setzt den Filter in der 1. Spalte auf alle Namen mit dem Vorzeichen B.

```
Sub Demo1 ()
    Sheets("Tabelle1").Cells(1, 1).AutoFilter _
        field:=1, _
        Criterial:="B*"
End Sub
```

Soll der Filter in einer Spalte wieder gelöscht werden, muss der Filter ohne Kriterium aufgerufen werden.

```
Sub Demo2 ()
    Sheets("Tabelle1").Cells(1, 1).AutoFilter _
        field:=1
End Sub
```

6 Datumswerte filtern

Name	Vorname	Geburtsdat
Müller	Maria	15.03.1982
Schmid	Anna	03.07.1974
Becker	Heike	08.03.1991
Schuster	Sonja	05.12.1981
Bauer	Zoe	05.06.1980
Schweitzer	Hans	07.07.1977
Bötcher	Lukas	05.05.1984
Brenner	Walter	06.09.1995
Schneider	Bernd	27.09.1990
Kutscher	Heiko	03.08.1985

Zum Filtern von Datumswerten werden diese im VBA-Standardformat erwartet. Dieses Format, das wir aus den USA kennen, hat die Form Monat/Tag/Jahr. Zur Umwandlung lässt sich die nachfolgende Funktion verwenden.

```
Function SystemDatumFormat(sDatum As String)
    SystemDatumFormat = Month(sDatum) & "/" & _
        & Day(sDatum) & "/" & Year(sDatum)
End Function
```

Damit funktioniert dann auch das Setzen eines Filters.

```
Sub Demo3()
    Dim sDatum As String
    sDatum = SystemDatumFormat("01.01.1988")
    Sheets("Tabelle1").Cells(1, 1).AutoFilter _
        field:=3, _
        Criteria1:="<" & sDatum
End Sub
```

Name	Vorname	Geburtsdat
Müller	Maria	15.03.1982
Schmid	Anna	03.07.1974
Schuster	Sonja	05.12.1981
Bauer	Zoe	05.06.1980
Schweitzer	Hans	07.07.1977
Bötcher	Lukas	05.05.1984
Kutscher	Heiko	03.08.1985

7 Dezimalzahlen filtern

Name	Vorname	Geburtsdatum	Reisekosten
Müller	Maria	15.03.1982	221,45
Schmid	Anna	03.07.1974	1.456,33
Becker	Heike	08.03.1991	2.555,10
Schuster	Sonja	05.12.1981	877,45
Bauer	Zoe	05.06.1980	1.236,55
Schweitzer	Hans	07.07.1977	1.110,98
Bötcher	Lukas	05.05.1984	556,00
Brenner	Walter	06.09.1995	3.145,76
Schneider	Bernd	27.09.1990	228,88
Kutscher	Heiko	03.08.1985	112,45

Bei Dezimalzahlen gibt es Probleme mit dem im Deutschen üblichen Dezimal-Komma. Am einfachsten ist es, das Komma gegen den Punkt zu tauschen und die nachfolgende Funktion zu nutzen.

```
Function SystemZahlFormat(sZahl As String)
    SystemZahlFormat = Application.Substitute(sZahl, ",", ".")
End Function
```

Damit funktioniert dann auch das Setzen eines Autofilters.

```
Sub Demo4()
    Dim sZahl As String
    sZahl = SystemZahlFormat("1000,00")
    Sheets("Tabelle1").Cells(1, 1).AutoFilter _
        field:=3, _
        Criteria1:=">=" & sZahl
End Sub
```

Name	Vorname	Geburtsdatum	Reisekosten
Schmid	Anna	03.07.1974	1.456,33
Becker	Heike	08.03.1991	2.555,10
Bauer	Zoe	05.06.1980	1.236,55
Schweitzer	Hans	07.07.1977	1.110,98
Brenner	Walter	06.09.1995	3.145,76

8 Texte mit Wildcards filtern

Datumswerte und Zahlen können nur komplett gesucht werden, bei Texten können die Wildcards „?“ und „*“ verwendet werden, um Zeilen herauszufiltern, deren Text dem eingegebenen Filterbegriff nur teilweise entspricht:

„Mü*“ findet zum Beispiel jeden Text, der mit den Buchstaben „Mü“ beginnt, „*Mü“ findet „Mü“ (auch kleingeschrieben) an jeder Stelle in einem Text. „Mü?“ findet jeden Text, der mit „Mü“ beginnt und dann noch nur einen dritten Buchstaben hat.