
HARALD NAHRSTEDT

Excel + VBA

Ergänzungen

Kapitel

Einführung in VBA

CustomViews in VBA nutzen

Erstellt am 15.10.2011

Beschreibung

In Excel besteht die Möglichkeit, benutzerdefinierten Ansicht zu erstellen. Damit lassen sich spezielle Ansichtseinstellungen (z. B. Spaltenbreite, Zeilenhöhe, ausgeblendete Zeilen und Spalten, Tabellenauswahl, Filtereinstellungen sowie Fenstereinstellungen) und Druckeinstellungen (z. B. Seiteneinstellungen, Ränder, Kopf- und Fußzeilen sowie Blatteinstellungen) für ein Arbeitsblatt speichern.

In Excel besteht die Möglichkeit, Benutzerdefinierte Ansicht zu erstellen. Damit lassen sich spezielle Ansichtseinstellungen (z. B. Spaltenbreite, Zeilenhöhe, ausgeblendete Zeilen und Spalten, Tabellenauswahl, Filtereinstellungen sowie Fenstereinstellungen) und Druckeinstellungen (z. B. Seiteneinstellungen, Ränder, Kopf- und Fußzeilen sowie Blatteinstellungen) für ein Arbeitsblatt speichern. Darin enthalten sein kann auch der Wechsel zwischen verschiedenen Tabellenblättern. In einer benutzerdefinierten Ansicht lassen sich ebenso spezielle Druckbereiche einbinden.

Pro Arbeitsblatt können Sie mehrere Benutzerdefinierte Ansichten erstellen. Eine Benutzerdefinierte Ansicht ist dem Workbook zugeordnet, in dem sie erstellt wurde, und nur in diesem verwendbar. Wenn Sie eine Benutzerdefinierte Ansicht nicht mehr benötigen, können Sie diese löschen.

Dieser Artikel befasst sich mit der Nutzung solcher Ansichten unter VBA.

1 Vorhandene Benutzerdefinierte Ansichten zeigen

Die nachfolgende Prozedur liest alle vorhandenen Ansichten aus dem aktiven Workbook und sammelt deren Namen.

```
Sub ShowAllCustomViews()  
    Dim objView As Object  
    Dim sText As String  
  
    sText = ""  
    For Each objView In ActiveWorkbook.CustomViews  
        sText = sText & objView.Name & vbLf  
    Next  
    MsgBox sText, vbOKOnly, "CUSTOMVIEWS"  
End Sub
```

Eine Benutzerdefinierte Ansicht lässt sich mit der nachfolgenden Prozedur aufrufen. Dabei wird zunächst der Name abgefragt.

```
Sub GoToCustomView()  
    Dim sName As String  
  
    On Error GoTo ErrInfo  
    sName = InputBox("Ansichtsname =", "CUSTOMVIEW")  
    ActiveWorkbook.CustomViews(sName).Show  
    Exit Sub  
ErrInfo:  
    MsgBox sName & " existiert nicht!", vbOKOnly & vbCritical, _  
        "F e h l e r !"  
End Sub
```

Mit einer Kombination aus beiden ließe sich eine Listbox füllen und die Anwahl daraus gestalten. Diese kleine Übung überlasse ich gerne dem Leser.

2 Eine Benutzerdefinierte Ansicht erstellen

Die nachfolgende Prozedur fragt nach einem Namen für die aktive Ansicht und speichert sie dann unter diesem Namen mit der Methode Add.

```
Sub CreateCustomView()  
    Dim sName As String  
  
    sName = InputBox("Ansichtsname =", "CUSTOMVIEW DEKLARATION")  
    ActiveWorkbook.CustomViews.Add Viewname:=sName, _  
        PrintSettings:=False, RowColSettings:=True  
End Sub
```

3 Eine Benutzerdefinierte Ansicht löschen

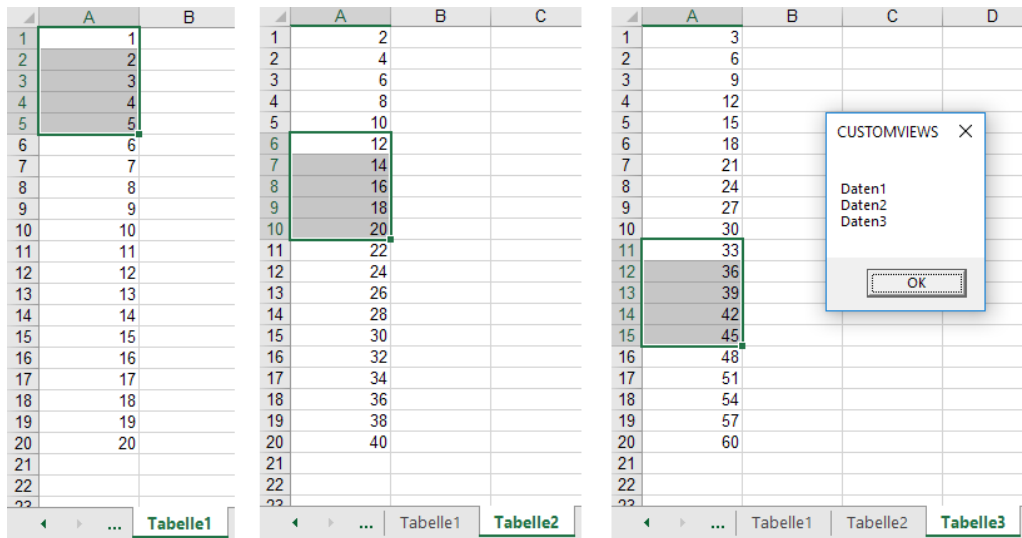
Die nachfolgende Prozedur fragt zuerst nach dem Namen der Ansicht, um danach diese mit der Methode Delete zu löschen.

```
Sub DeleteCustomView()  
    Dim sName As String  
  
    sName = InputBox("Ansichtsname =", "CUSTOMVIEW DELETE")  
    On Error GoTo ErrInfo:  
    ActiveWorkbook.CustomViews(sName).Delete  
    MsgBox sName & " gelöscht!", _  
        vbOKOnly & vbInformation, "Bestätigung"  
    Exit Sub  
ErrInfo:  
    MsgBox sName & " nicht gelöscht!", _  
        vbOKOnly & vbCritical, "F e h l e r !"  
End Sub
```

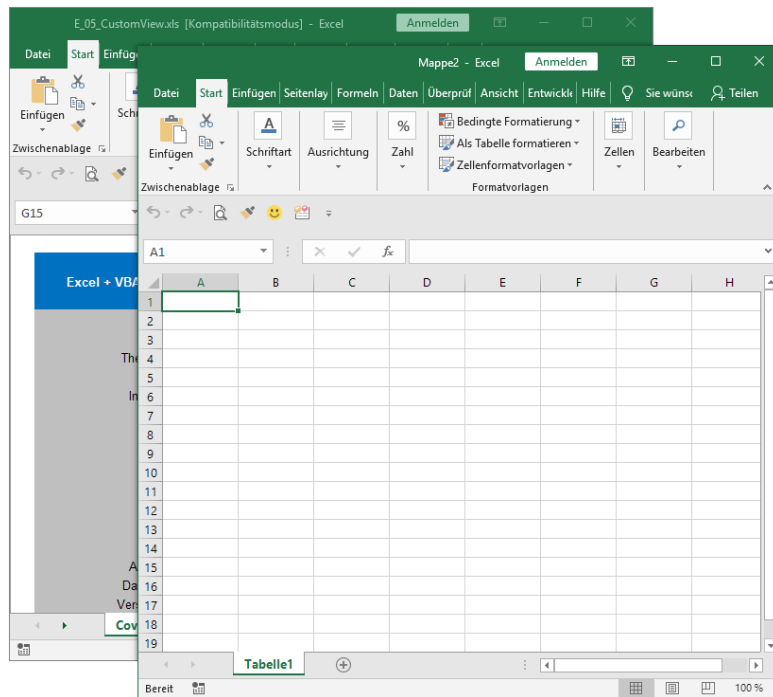
4 Benutzerdefinierte Ansichten übertragen

Etwas mehr Programmieraufwand kostet es, will man die benutzerdefinierten Ansichten einer Mappe auf eine andere übertragen.

Zur Erklärung benutzen wir ein kleines Beispiel. In drei Tabellen schreiben wir die dargestellten Zahlen. Dann markieren wir in Tabelle1 den Bereich A1:A5 und erstellen die Benutzerdefinierte Ansicht *Daten1*. In Tabelle2 markieren wir den Bereich A6:A10 und erstellen die Benutzerdefinierte Ansicht *Daten2*. In Tabelle3 markieren wir den Bereich A11:A15 und erstellen die Benutzerdefinierte Ansicht *Daten3*.



Nun wollen wir diese Benutzerdefinierten Ansichten auf eine andere Mappe übertragen. Dazu öffnen wir eine weitere leere Excel-Anwendung.



Gestartet wird die nachfolgende Prozedur in der Mappe mit den benutzerdefinierten Ansichten.

```
Sub CopyCustomView()  
    Dim AltBook As Workbook  
    Dim NeuBook As Workbook  
    Dim Finde As Boolean  
    Dim cmvAlt As CustomView  
    Dim cmvNeu As CustomView  
    Dim sAdd As String  
  
    Set AltBook = ThisWorkbook ' in alter Mappe starten  
  
    ' Durchlaufe alle Mappen  
    For Each NeuBook In Workbooks  
        If NeuBook.Name <> AltBook.Name And _  
            NeuBook.Windows(1).Visible Then _  
            ' neue Mappe gefunden  
            Finde = True  
            Exit For  
        End If  
    Next  
  
    ' Andere Mappe nicht gefunden?  
    If Not Finde Then  
        MsgBox "Bitte 2te Mappe in der gleichen Anwendung öffnen!"  
        Exit Sub  
    End If  
  
    ' Zuerst in der neuen Mappe die Standardansicht speichern  
    NeuBook.Activate ' neue Mappe aktiv  
    Set cmvNeu = _  
        NeuBook.CustomViews.Add("Standardansicht", True, True)  
  
    ' Durchlaufe alle Ansichten in der alten Mappe  
    For Each cmvAlt In AltBook.CustomViews  
        AltBook.Activate ' alte Mappe aktiv  
        cmvAlt.Show ' Ansicht anzeigen  
        sAdd = Selection.Address(0, 0) ' Selektierte Zellen merken  
        Cells.Copy ' Zellen kopieren  
        NeuBook.Activate ' Die andere Mappe aktivieren  
        cmvNeu.Show ' Standardansicht anzeigen  
        Cells.PasteSpecial Paste:=xlPasteFormats 'Formate einfügen  
        Range(sAdd).Select ' Zellen selektieren  
        ' Ansicht mit gleichem Namen speichern  
        NeuBook.CustomViews.Add cmvAlt.Name, _  
            cmvAlt.PrintSettings, cmvAlt.RowColSettings  
    Next  
  
    'Kopiermodus aus  
    Application.CutCopyMode = False  
End Sub
```

Danach befinden sich alle benutzerdefinierten Ansichten auch auf der neuen Mappe.

