Eine eigene Zugangsverwaltung für Excel

Bernd Held, Vaihingen an der Enz

Inhalt

Standardmäßig können Sie Excel-Arbeitsmappen individuell über einen Kennwortschutz schützen und somit eine kleine Benutzerverwaltung definieren. Möchten Sie nun aber eine anwenderspezifische Benutzerverwaltung einrichten, so ist dies über den Einsatz einiger Makros und Ereignissen in Excel möglich, eine zuverlässige Benutzerverwaltung erstellen. Dabei wird beim Öffnen einer bestimmten Excel-Arbeitsmappe geprüft, welcher Anwender aktuell angemeldet ist. Je nach Anwenderstatus können dann unterschiedliche Aufgaben freigegeben beziehungsweise gesperrt werden.

Alle Beispiele dieses Artikels finden Sie in der Demodatei Zugangsschutz.xls.

1 Benutzerverwaltung einrichten

Stellen Sie sich vor, es besteht die Anforderung bereits beim Öffnen einer Datei, abhängig vom Nutzer Tabellenblätter anzuzeigen oder auch auszublenden. Oder es besteht der Wunsch nutzerabhängig „Aufenthaltsbereiche“ innerhalb einer Tabelle zu definieren. Für diese und auch andere ähnliche Anforderungen muss bereits beim Öffnen der Datei ermittelt werden, wer diese Arbeitsmappe geöffnet hat. Durch die Abfrage des bei Windows festgelegten Anwendernamens kann dann je nach dem entschieden werden, wie weiter vorgegangen wird.

Damit die Überprüfung des Anwenders direkt beim Öffnen der Arbeitsmappe durchgeführt wird, setzen Sie das Arbeitsmappenereignis `Workbook_Open` ein. Dazu wechseln Sie über die Tastenkombination Alt + F11 in die Entwicklungsumgebung von Excel und führen einen Doppelklick auf den Eintrag DieseArbeitsmappe im Projekt-Explorer durch. Stellen Sie im ersten Dropdown (links oben) den Eintrag Workbook ein. Dadurch öffnet sich ein leerer Ereignisrahmen, in dem nun das folgende Listing übernommen werden kann.

```
Private Sub Workbook_Open()
Dim strID As String
Dim intAnw As Integer
Dim TblAnw As Worksheet

strID = Environ("username")

Set TblAnw = Worksheets("Anwender")
With TblAnw
For intAnw = 2 To .UsedRange.Rows.Count
 If LCase(.Cells(intAnw, 1).Value) = LCase(strID) Then
  Select Case .Cells(intAnw, 3).Value
    Case "high"
      JedeTabelleEinblenden
      TabellenschutzAus
    Case "standard"
      NurBestimmteTabellenEinblenden
      MitScrollArea
      TabellenschutzEin
    Case "low"
      NurEineTabelleEinblenden
```

```
      MitScrollArea
      TabellenschutzEin
    Case Else

  End Select
 Exit For
 End If
Next intAnw
End With
Worksheets("Start").Activate
End Sub
```

Nach einer erfolgreichen Ermittlung des Anwendernamens über die Anweisung `Environ("username")` wird dieser nun über eine `For Next`-Schleife mit dem Namen aus der Tabelle „Anwender“ (siehe Abb.1) verglichen. In dieser Tabelle befindet sich in Spalte A der tatsächliche Anwendernamen und zusätzlich in Spalte B auch noch der „Real-Name“ als zusätzliche Info. In Spalte C ist festgelegt, zu welcher „Zugriffsgruppe“ der jeweilige Anwender gehört.

Beim Abgleich der Variablen `StrID`, in der der ermittelte Anmeldenamen steht, mit der jeweiligen Zelle in Spalte A werden beide Informationen über die Standard-VBA-Funktion `LCase` zum identischen Abgleich in Kleinbuchstaben umgewandelt. Dies ist notwendig, da die `If`-Anweisung zwischen Groß- und Kleinschreibung unterscheidet. Über diesen Trick können eventuelle „Ungenauigkeiten“ bei der Pflege der Tabelle „Anwender“ ausgeglichen werden.

	A	B	C	D	E
1	**Anmeldename**	**Name**	**Gruppe**		
2	Held	Held	high		
3	Mueller	Müller	high		
4	Schmitt	Schmitt	standard		
5	Postner	Postner	standard		
6	Pippig	Pippig	low		
7	Buerg	Bürg	low		
8					
9					

Start | Anwender

Abb. 1: <Die zugelassenen Anwender

Innerhalb einer Schleife werten Sie die Spalte C der Tabelle „Anwender“ aus. Mögliche Werte sind dabei:

- High: Bei dieser Einstellung darf der so gekennzeichnete Anwender alle Tabelle einsehen und verändern. Zu diesem Zweck werden über das Makro `JedeTabelleEinblenden` alle standardmäßig ausgeblendeten Tabellen eingeblendet sowie über das Makro TabellenschutzAus alle eingestellten Schutzmechanismen deaktiviert.
- Standard: Bei dieser Sicherheitseinstellung darf der Anwender nur Zugriff auf bestimmte Tabellen haben, welches das Makro `NurBestimmteTabellenEinlenden` sicherstellt. In diesen Tabellen darf der so gekennzeichnete Anwender sich lediglich in einem vorgegebenen Bereich aufhalten. Diese Aufgabe wird über das Makro `MitScrollArea` gelöst. Des Weiteren haben Anwender dieser Sicherheitsstufe keine Möglichkeit, Werte in den Tabellen zu ändern beziehungsweise zu löschen. Zu diesem Zweck wird das Makro `TabellenschutzEin` genutzt.

- Low: Bei dieser Sicherheitseinstellung darf der Anwender nur noch die Tabelle „Start“ einsehen.

2 Die Schutz-Makros

Auf den folgenden Seiten sehen Sie nun die einzelnen Funktionen des Zugangsschutzes.

2.1 Tabellen ein- und ausblenden

Um Tabellen je nach ermitteltem Anwenderstatus ein- beziehungsweise auszublenden, setzen Sie die folgenden Makros ein.

```
Sub JedeTabelleEinblenden()
Dim tbl As Worksheet

For Each tbl In ThisWorkbook.Worksheets
 tbl.Visible = True
Next tbl
End Sub

Sub NurBestimmteTabellenEinblenden()
 Worksheets("Tabelle1").Visible = True
 Worksheets("Tabelle2").Visible = True
 Worksheets("Tabelle3").Visible = True
End Sub

Sub AlleTabellenAusserStartAusblenden()
Dim tbl As Worksheet

For Each tbl In ThisWorkbook.Worksheets
 If tbl.Name <> "Start" Then
  tbl.Visible = xlSheetVeryHidden
 End If
Next tbl
End Sub

Sub NurEineTabelleEinblenden()
 Sheets("Tabelle1").Visible = True
End Sub
```

Um Tabellen sicher auszublenden, setzen Sie die Eigenschaft `Visible` der jeweiligen Tabelle auf den Wert `xlsheetVeryHidden`. Diese Methode bewirkt, dass der „Standard Anwender“ keine Möglichkeit besitzt, die sicher ausgeblendeten Tabellen über die Standardoberfläche und dem Menübefehl Format/Blatt/einblenden wieder sichtbar zu machen. Dieser Menübefehl ist für sicher ausgeblendete Tabellen deaktiviert.

2.2 Die ScrollArea einrichten

Unter einer SCROLLAREA versteht man in Excel einen zusammenhängenden definierten Bereich in einer Excel-Tabelle, indem sich ein Anwender bewegen darf. Dieser Bereich ist standardmäßig in Excel nicht gesetzt, was bedeutet, dass ein Anwender Zugriff auf alle 256 Spalten und 65536 Zeilen einer Tabelle hat. Dieser Bereich kann deutlich eingeschränkt werden. So wird die ScrollArea im folgenden Makro bei allen momentan sichtbaren Tabellen für den Bereich A1:E20 festgesetzt. Es ist nach der Festlegung der ScrollArea nicht mehr möglich, Eingaben außerhalb des Bereiches vorzunehmen oder Zellen außerhalb dieses Bereiches anzusteuern.

```
Sub MitScrollArea()
Dim tbl As Worksheet

For Each tbl In ThisWorkbook.Worksheets
 If tbl.Visible = True Then
   tbl.ScrollArea = "A1:E20"
 End If
Next tbl
End Sub
```

Deklarieren Sie im ersten Schritt des Makros eine Objektvariable vom Typ `Worksheet`. Damit erhält man Zugriff auf alle Methoden und Eigenschaften, die für Tabellen verfügbar sind. Über eine `For Each Next`-Schleife werden alle Tabellen der Arbeitsmappe „durchlaufen. Innerhalb der Schleife wird überprüft, ob die jeweilige Tabelle eingeblendet ist. In diesem Fall wird mit der Eigenschaft `ScrollArea` der „Aufenthaltsbereich“ des Anwenders festgelegt. Übergeben Sie dieser Eigenschaft die exakten Koordinaten des Bereichs der Tabelle, welcher als ScrollArea definiert werden soll.

2.3 Tabellenschutz an und ausschalten

Als zusätzliches Feature soll allen Anwender mit dem Benutzerprofil high die Möglichkeit gegeben werden Änderungen in eine Tabelle durchzuführen, ohne dabei jedes Mal das Kennwort des Tabellenschutzes eingeben zu müssen. Hierfür muss für Anwender der Sicherheitsstufe high generell der Passwortschutz aller Tabellen entfernt werden.

Im folgenden Makro wird der Tabellenschutz auf allen Tabellen der Arbeitsmappe entfernt.

```
Sub TabellenschutzAus()
Dim tbl As Worksheet

For Each tbl In ThisWorkbook.Worksheets
 tbl.Unprotect Password:="Test"
Next tbl
End Sub
```

In einer `For Each-Next`-Schleife wird über die Methode `Unprotect` der Tabellenschutz deaktiviert. Dabei übergeben Sie im Argument `Password` das Kennwort für den Tabellenschutz.

Beim Einstellen des Tabellenschutzes stehen standardmäßig bestimmte Funktionen wie das Filtern von Daten sowie die Gruppierungs- und Gliederungsfunktionen in Excel leider nicht mehr zur Verfügung. Ab der Excel-Version 2002 können diese Funktionen selektiv trotz eingestelltem Tabellenschutz durchgeführt werden. So kann beim Schützen einer Tabelle genau über den Dialog Blatt schützen festgelegt werden, welche Aktion ein Anwender auch in einer geschützten Tabelle durchführen darf.

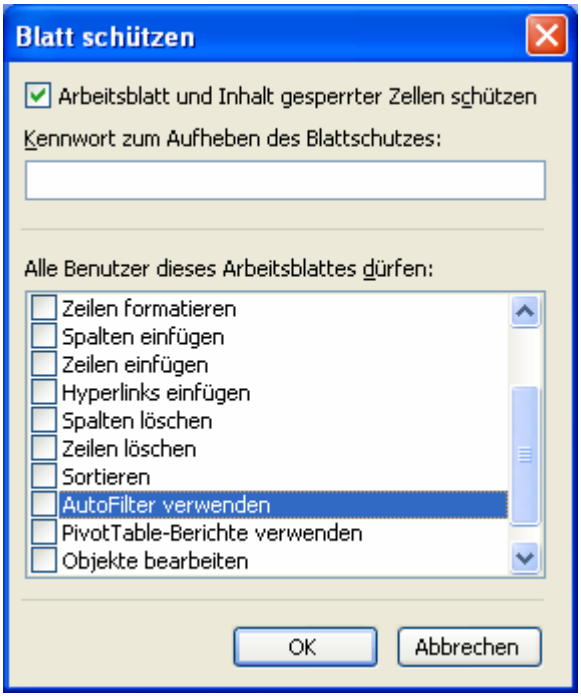

Abb. 2: <Tabellenschutz einstellen>

Anwender für die Excel-Versionen 97-2000 kann aber dennoch geholfen werden! Aktivieren Sie vor dem eigentlichen Schutz einen Autofilter und blenden die Gruppierungssymbole über die Funktion Teilergebnis ein Mit dem „Trick“ aus dem folgenden Listing wird erreicht, dass diese beiden Funktionen auch nach dem Einstellen des Tabellenschutzes noch funktionieren. Dieses Makro ist für die Versionen Excel 97-2000 sowie für die neuen Versionen Excel 2002-2003 universal einsetzbar.

```
Sub TabellenschutzEin()
Dim tbl As Worksheet

For Each tbl In ThisWorkbook.Worksheets
 tbl.EnableAutoFilter = True
 tbl.EnableOutlining = True
 tbl.Protect Password:="Test", userinterfaceonly:=True
Next tbl
End Sub
```

Über eine `For Each-Next`-Schleife werden alle Tabellen der Arbeitsmappe angesteuert. Innerhalb der Schleife wird die Eigenschaft `EnableAutoFilter` sowie die Eigenschaft `EnableOutlining` auf den Wert `True` gesetzt, um die Filterung und Bedienung der Gliederungssymbole sicherzustellen. Wenden Sie danach die Methode `Protect` an, übergeben das gewünschte Kennwort für den Tabellenschutz. Die gewünschte Funktionalität wird über das Argument `userinterfaceonly` und dem Wert `True` aktiviert.

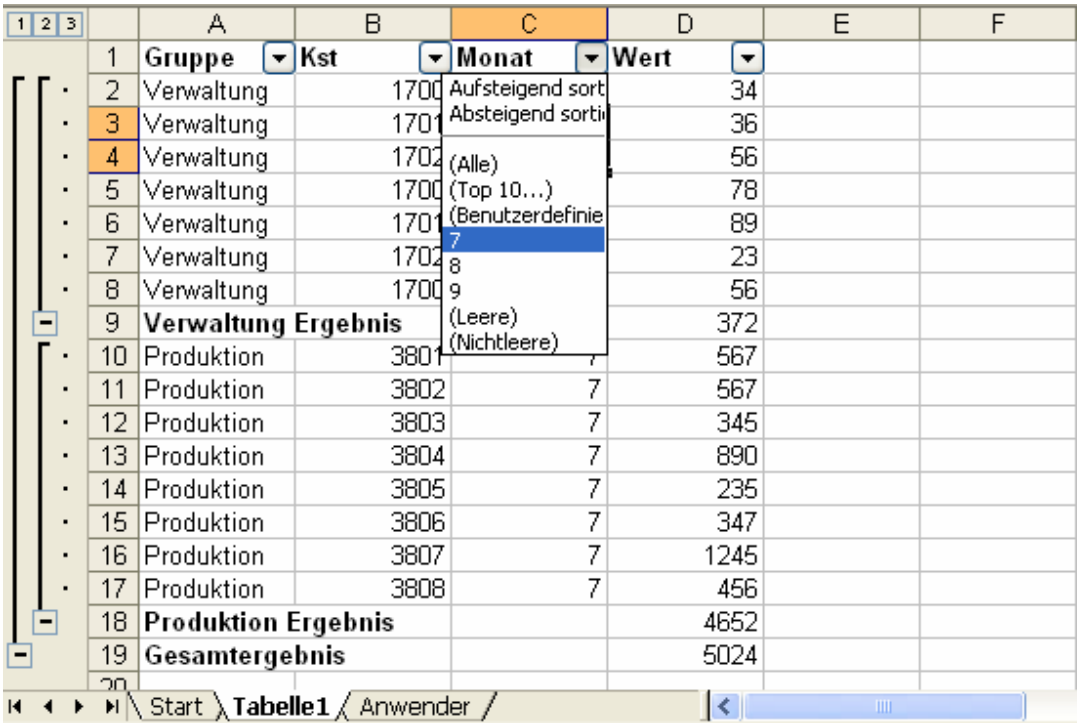

Abb. 3: <Filtern trotz Blattschutz ermöglichen>

3 Bestimmte Aktionen verhindern

Je nach Gruppe können Sie auch bestimmte Aktionen wie zum Beispiel Drag&Drop verhindern bzw. auch Tastenkombinationen oder bestimmte Menüs deaktivieren.

3.1 Drag & Drop abschalten

Soll das Ziehen und Ablegen von Zellen auf einer Tabelle verhindert werden, um das versehentliche Überschrieben von Daten zu verhindern, dann können Sie hierzu das folgende Makro einsetzen.

```
Sub DragDropDeaktivieren()
 Application.CellDragAndDrop = False
End Sub

Sub DragDropAktivieren()
 Application.CellDragAndDrop = True
End Sub
```

Schieben Sie diese Makros wahlweise je nachdem für welche Gruppen Sie die Funktion haben möchten in den entsprechenden `Case`-Zweig im Ereignis `Workbook_Open`.

3.2 Menübefehl(e) deaktivieren

Um einen speziellen Menübefehl wie beispielsweise den Befehl Optionen aus dem Menü Extras zu deaktivieren, um Änderungen an den allgemeinen Einstellungen zu verhindern, können Sie das folgende Makro einsetzen.

```
Sub MenueExtrasOptionenDEaktivieren()
Application.CommandBars(1).Controls _
("E&xtras").Controls("&Optionen...").Enabled = False
End Sub
```

Wird dieses Makro ausgeführt, ist der Menübefehl Extras → Optionen gegraut und kann nicht ausgewählt werden. Starten Sie das folgende Makro, um diesen Befehl wieder anklickbar zu machen.

```
Sub MenueExtrasOptionenAktivieren()

 Application.CommandBars(1).Controls _
("E&xtras").Controls("&Optionen...").Enabled = True

End Sub
```

3.3 Tastenkombinationen ausschalten

Sollen beispielsweise die Tastenkombinationen Strg + C und Strg + V für das Kopieren und Einfügen von Daten ausgeschaltet werden, dann setzen Sie das folgende Makro ein.

```
Sub TastenkombinationenAusschalten()

 Application.OnKey "^c", ""
 Application.OnKey "^v", ""

End Sub

Sub TastenkombinationenEinschalten()

 Application.OnKey "^c"
 Application.OnKey "^v"

End Sub
```

Das Caret-Zeichen steh für die Taste Strg. Wird im zweiten Argument der Methode `OnKey` eine leere Zeichenfolge angegeben, dann wird die bisherig zugeordnete Funktion damit deaktiviert. Soll die orginäre Funktion der Tastenkombination wieder verfügbar gemacht werden, dann rufen Sie die Methode `Onkey` ohne zweites Argument auf.

4 Der Abschluss

Die hier gezeigten Beispiele sind nur dann sicher, wenn Sie nach dem Einstellen der gewünschten Funktionen den Zugang zum Quellcode verhindern. Dazu müssen Sie das Projekt wie folgt schützen:

1 Wechseln Sie über die Tastenkombination Alt + F11 in die Entwicklungsumgebung von Excel.

2 Wählen Sie aus dem Menü Extras den Befehl Eigenschaften von VBAProjekt.

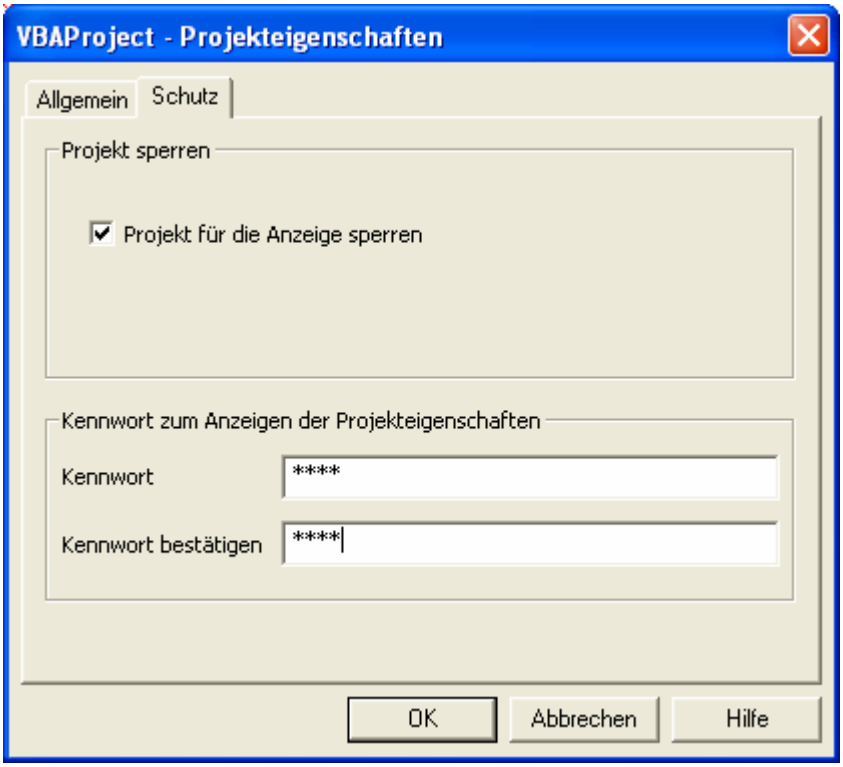

Abb. 4: <Makros und Passwörter schützen>

3 Aktivieren Sie das Kontrollkästchen Projekt für die Anzeige sperren.

4 Vergeben Sie ein Kennwort und bestätigen es.

5 Klicken Sie OK.

6 Speichern Sie die Arbeitsmappe.

Dieser Schutz wird wirksam, wenn Sie die Mappe schließen und erneut öffnen. Danach können Sie den Quellcode nur einsehen, wenn Sie vorher das richtige Kennwort eingegeben haben.