

# Alternative Schutzmechanismen für Excel

Bernd Held, Vaihingen an der Enz

Neben dem in Excel standardmäßig angebotene Tabellenschutz können Sie weitere Schutzmechanismen in Excel einsetzen. Dazu bedarf es nur weniger Zeilen Makrocodes. Dieser Artikel stellt Ihnen ein paar dieser Schutzmechanismen vor. Trotzdem: Den hundertprozentigen Schutz Ihrer Daten wird es meiner Ansicht nach nie geben können. Sie können es aber „Hackern“ so schwer wie möglich machen, indem Sie mehrere Schutzmechanismen parallel einsetzen.

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

## Inhalt

- |                                      |
|--------------------------------------|
| 1 Eine ScrollArea definieren.....1   |
| 2 Auswahl von Bereichen verhindern3  |
| 3 Eingaben in Bereichen widerrufen.4 |
| 4 Geänderte Daten nicht speichern..5 |



## 1 Eine ScrollArea definieren

Ein etwas anderer Ansatz als das komplette Schützen von Tabellen ist, bestimmte Bereiche schon gar nicht für Anwender zugänglich machen. In Excel können Sie mit ein paar Angriffen eine Scrollarea einstellen. Damit ist ein Bereich gemeint, in dem sich der Anwender in einer Tabelle aufhalten darf. In diesem Bereich kann der Anwender dann machen, was er will – allen Zellen außerhalb der Scrollarea sind unantastbar. Um beispielsweise für die Tabelle1 die Scrollarea A1:D10 zu definieren, gehen Sie wie folgt vor:

- 1 rücken Sie die Tastenkombination Alt + F11, um in die Entwicklungsumgebung von Excel zu gelangen.
- 2 Im Projekt-Explorer öffnen Sie die Rubrik Microsoft Excel Objekte und führen einen Doppelklick auf den Eintrag DieseArbeitsmappe durch.
- 3 Wählen Sie im Codefenster aus dem Dropdown oben links das Ereignis Open aus.
- 4 Ergänzen Sie den noch leeren Ereignisrahmen wie folgt:

```
Private Sub Workbook_Open()  
    Tabelle1.ScrollArea = "A1:D10"  
End Sub
```

Das Ereignis Workbook\_Open wird automatisch beim Öffnen der Arbeitsmappe ausgeführt. Dabei wird dann die ScrollArea für Tabelle1 eingestellt. Dem „normalen“ Anwender ist es danach nicht möglich, Zellen außerhalb dieses Bereiches anzusteuern.

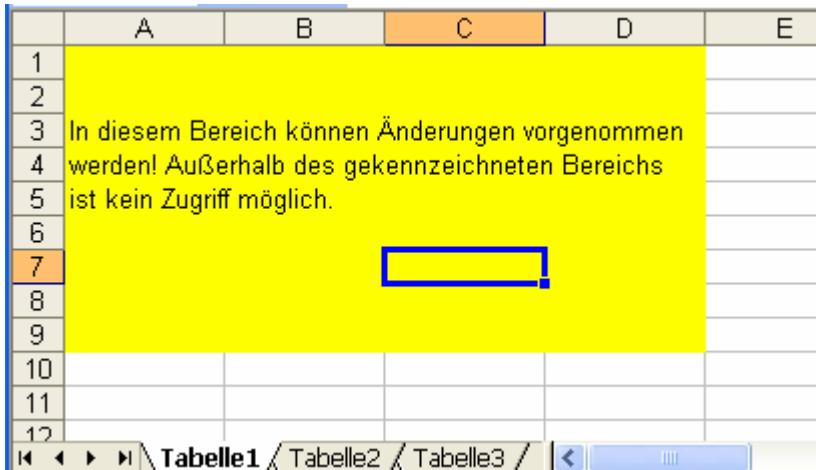


Abb. 1: <Die Scrollarea wird beim Öffnen der Mappe festgelegt>

### 1.1 Erweiterung der ScrollArea

Wenn Sie möchten, können Sie die Einstellung der ScrollArea auch abhängig vom Anwender machen, der auf die Arbeitsmappe zugreift. Erweitern Sie dazu das vorherige Beispiel wie folgt:

```
Private Sub Workbook_Open()

Select Case Environ("username")
Case "Held", "Meier", "Schmidt"
'keine ScrollArea einstellen
Tabelle1.ScrollArea = ""

Case Else
Tabelle1.ScrollArea = "A1:D10"

End Select

End Sub
```

Fügen Sie bei Bedarf weitere Anwender, für die keine Scrollarea eingestellt werden soll, beim ersten Case-Zweig jeweils getrennt durch ein Komma in doppelten Anführungszeichen ein. Mit dem Befehl Environ("username") können Sie den Anmeldenamen des Anwenders unter Windows spielend auslesen und danach über eine Select Case Anweisung auswerten.



Ob für eine Tabelle eine Scrollarea eingestellt ist, können Sie überprüfen, indem Sie in der Entwicklungsumgebung das Eigenschaften-Fenster einblenden und die jeweilige Tabelle im Projekt-Explorer markieren.

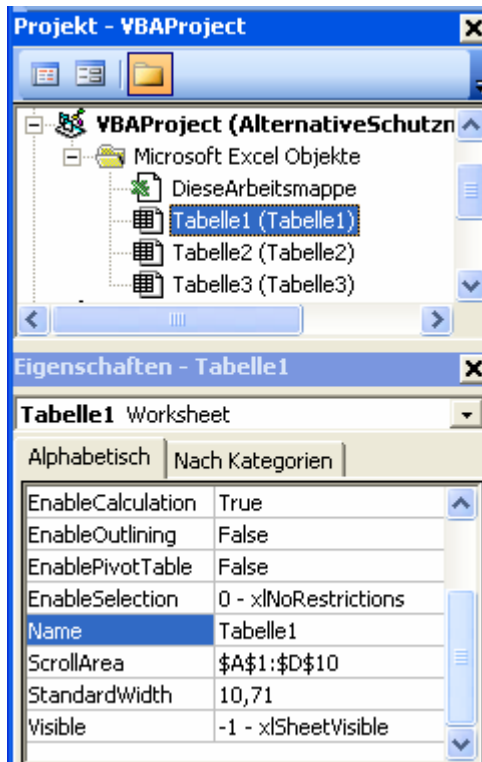


Abb. 2: &lt;Eine ScrollArea entfernen&gt;

Wenn Sie im Feld ScrollArea den Zellenbezug entfernen, ist die Tabelle wieder frei editierbar. Dies funktioniert natürlich nur, wenn der Zugang zum Quellcode nicht durch ein Passwort gesichert ist. Sämtlicher Quellcode kann dem normalen Anwender versperrt werden. Dazu wird im Projekt-Explorer bei der Arbeitsmappe mit der rechten Maustaste geklickt und der Befehl Eigenschaften von VBAProjekt ausgewählt. Auf der Registerkarte Schutz wird danach das Kontrollkästchen Projekt für die Anzeige sperren aktiviert sowie ein Passwort eingegeben und bestätigt. Nach der Speicherung kann der Quellcode der Arbeitsmappe nach erneutem Öffnen derselben nicht ohne Kenntnis des Passworts zur Ansicht gebracht werden. Somit ist auch die eingestellte ScrollArea nicht sichtbar.

Ein kleiner Nachteil der Scrollarea ist, dass es sich dabei um einen zusammenhängenden Bereich handeln muss und es nur eine einzige ScrollArea pro Tabelle geben darf. Dieses kleine Manko können Sie mit der folgenden Lösung ausschalten.

## 2 Auswahl von Bereichen verhindern

Bei der folgenden praktischen Lösung wird das Markieren bestimmter Zellen insoweit verhindert, dass nach der Auswahl dieser Zellen automatisch der Mauszeiger in eine andere Zelle gesetzt wird. Dieser Trick sorgt dafür, dass bestimmte Zellen eben auch ohne eingestellten Tabellenschutz nicht editierbar sind, da sie nicht anwählbar sind.

Wenden Sie dieses Verhalten für die Tabelle2 für die Bereiche A1:B5 und D1:D5 an, indem Sie die nächsten Arbeitsschritte befolgen:

- 5 Klicken Sie mit der rechten Maustaste den Tabellenreiter (unten) der Tabelle2 an.
- 6 Wählen Sie aus dem Kontextmenü den Befehl Code anzeigen, um direkt in die Entwicklungsumgebung zu wechseln.
- 7 Stellen Sie im Dropdown-Feld des Code-Fensters den Eintrag Worksheet ein. Damit wird das Ereignis Worksheet\_SelectionChange ein, das automatisch bei jedem Versetzen des Mauszeigers ausgeführt wird.



## 8 Erweitern Sie den noch leeren Ereignisrahmen wie folgt:

```
Private Sub Worksheet_SelectionChange _
    (ByVal Target As Range)

    Dim Bereich1 As Range
    Dim Bereich2 As Range
    Dim Gesamt As Range

    Set Bereich1 = Tabelle2.Range("A1:B5")
    Set Bereich2 = Tabelle2.Range("D1:D5")
    Set Gesamt = Union(Bereich1, Bereich2)

    If Intersect(Gesamt, Target) Is Nothing Then

    Else
        Tabelle2.Range("C7").Select
    End If

End Sub
```

Wird nun versucht eine Zelle in den Bereichen A1:B5 bzw. D1:D5 anzusteuern, wird der Mauszeiger augenblicklich in Zelle C7 verschoben. Nun ein paar Sätze zum Quellcode. Zuerst werden einige Bereichsvariablen definiert, die dann über die Anweisung Set näher bestimmt werden. Über die Anweisung Union können Sie diese Einzelbereiche zu einem Gesamtbereich zusammenlegen. Bei diesem Bereich müssen die einzelnen Bereiche nicht nebeneinander liegen. Über den Befehl Intersect können Sie überprüfen, ob die momentan markierte Zelle (Target) in dem definierten Gesamtbereich liegt. Wenn ja, dann setzen Sie den Mauszeiger über den Befehl Select in Zelle C7.

	A	B	C	D	E
1	Diese Zellen dürfen nicht			editiert werden!	
2					
3					
4					
5					
6					
7					

Abb. 3: <Die gekennzeichneten Bereiche sind nicht zugänglich>

## 3 Eingaben in Bereichen widerrufen

Auch eine gängige Variante bestimmte Bereiche zu schützen ist, darin vorgenommene Eingaben sofort wieder zu widerrufen. Im folgenden Beispiel soll in Tabelle3 der Bereich A1:D10 auf diese Art und Weise geschützt werden. Gehen Sie dabei wie folgt vor:



- 1 Klicken Sie mit der rechten Maustaste den Tabellenreiter (unten) der Tabelle2 an.
- 2 Wählen Sie aus dem Kontextmenü den Befehl Code anzeigen, um direkt in die Entwicklungsumgebung zu wechseln.

- 3 Stellen Sie im linken Dropdown-Feld des Code-Fensters den Eintrag Worksheet ein.
- 4 Wählen Sie im rechten Dropdown-Feld das Ereignis Change ein. Dieses Ereignis wird bei jeder Eingabe in eine beliebige Zelle der Tabelle ausgeführt.
- 5 Erweitern Sie den noch leeren Ereignisrahmen wie folgt:

```
Private Sub Worksheet_Change _
    (ByVal Target As Range)

    Dim Bereich As Range
    Set Bereich = Tabelle3.Range("A1:D10")

    If Intersect(Target, Bereich) Is Nothing Then

    Else
        Application.EnableEvents = False
        MsgBox "Eingabe in diesem Bereich verboten!" & _
            vbCrLf & "Eingabe wird rückgängig gemacht!", _
            vbCritical
        Application.Undo
        Application.EnableEvents = True
    End If

End Sub
```

Nach dem Definieren und Festlegen des Bereichs, der geschützt werden soll, wird über die Anweisung Intersect geprüft, ob eine Eingabe im definierten Bereich stattgefunden hat. Wenn ja, dann wird die Ereignissteuerung von Excel kurz ausgeschaltet, die Eingabe über den Befehl Undo widerrufen und anschließend die Ereignissteuerung wieder eingeschaltet.



Abb. 4: <Eingabe sofort wieder rückgängig machen>

#### 4 Geänderte Daten nicht speichern

Etwas „gemein“ ist auch die folgende Variante. Dabei lassen Sie den Anwender Daten in Ihrer Mappe zwar ändern, wenn er diese Änderungen aber speichern möchte, lassen Sie dieses nur zu, wenn er ein Passwort eingibt.

Stellen Sie dieses Feature wie folgt ein:



- 1 Drücken Sie die Tastenkombination Alt + F11, um in die Entwicklungsumgebung von Excel zu gelangen.
- 2 Im Projekt-Explorer öffnen Sie die Rubrik Microsoft Excel Objekte und führen einen Doppelklick auf den Eintrag DieseArbeitsmappe durch.
- 3 Wählen Sie im Codefenster aus dem Dropdown oben links das Ereignis BeforeSave aus.
- 4 Ergänzen Sie den noch leeren Ereignisrahmen wie folgt:

```
Private Sub Workbook_BeforeSave _  
    (ByVal SaveAsUI As Boolean, Cancel As Boolean)  
  
    Dim strText As String  
  
    strText = InputBox("Bitte Passwort eingeben!")  
  
    If strText <> "Test" Then  
        MsgBox "Sie sind unbefugt Änderungen abzuspeichern" & _  
            vbLf & "Ihre Änderungen werden nicht gesichert!", _  
            vbExclamation  
        Cancel = True  
    End If  
  
End Sub
```

Das Ereignis Workbook\_BeforeSave wird automatisch beim Speichern der Arbeitsmappe ausgeführt. Dies ist der ideale Zeitpunkt, um dazwischen zu funken. Fragen Sie dort ein Passwort ab. Wird dieses Passwort nicht eingegeben brechen Sie den kompletten Speichervorgang ab, indem Sie das Argument Cancel auf den Wert True setzen.