

Suchen und ersetzen mit externer Tabelle und Benutzerdialog

Dr. Volker Thormählen, 10. Mai 2010

Inhalt

1	Aufgabe	2
2	Referenzdokument.....	2
3	Zieldokument	4
4	Prozeduren	5
5	Prozeduren und Funktionen im Modul " <i>ReplaceFromTable</i> "	6
6	VBA-Quellcode der Anwendung.....	6

Abbildungen

Abbildung 1: Projekt "Replacer"	5
Abbildung 2: Struktogramm der Anwendung "ReplaceFromTable"	6

Tabellen

Tabelle 1: Suchbegriffe und Ersetzungen	4
Tabelle 2: Objekte der Anwendung.....	6

1 Aufgabe

In diesem Beitrag wird beschrieben, wie ein ausgewähltes Word-Dokument (sog. Zieldokument) bereinigt werden kann, indem die eingebaute Word-Funktionalität zum Suchen und Ersetzen im Rahmen von VBA-Prozedur und -Funktionen genutzt wird. Fundstellen im Zieldokument werden nur dann ersetzt, wenn der Benutzer der vorgeschlagenen Ersetzung zustimmt. Wonach im Zieldokument gesucht und ersetzt wird, ergibt sich aus einer einschlägigen Tabelle in einem Referenzdokument.

2 Referenzdokument

Das Referenzdokument mit dem Namen "*CheckDocuments*" enthält eine Tabelle mit 59 Zeilen und 3 Spalten (s. Tabelle 1). Die erste Zeile der Tabelle enthält die Spaltenköpfe. Sie lauten:

- Suchen nach ...
- Ersetzen durch ...
- Beschreibung

Suchen nach ...	Ersetzen durch ...	Beschreibung
(,)([.,;:^0130^0132^0145^0147\?!\])	\2	Komma vor best. Satzzeichen o. linkem Anführungszeichen: Entfernen!
(.)([.,;:^0130^0132^0145^0147\?!\])	\2	Punkt vor best. Satzzeichen o. linkem Anführungszeichen: Entfernen!
(:)([.,;:^0130^0132^0145^0147\?!\])	\2	Doppelpunkt vor best. Satzzeichen o. linkem Anführungszeichen: Entfernen
(;)([.,;:^0130^0132^0145^0147\?!\])	\2	Semikolon vor best. Satzzeichen o. linkem Anführungszeichen: Entfernen!
([]){2;}	\1	2 oder mehr ungeschützte Leerzeichen in Folge
([]{1;})([.,;:\?!\^0130^0132^0145^0147\)\])	\2	1 oder mehr ungeschützte Leerzeichen vor Interpunktionszeichen bzw. öffnenden typografischen Anführungszeichen
([!])^0133	\1^0133	Fehlendes Leerzeichen vor horizontaler Ellipse: Einfügen!
([.,;:\?!\)\])}{2;}	\1	2 oder mehr Interpunktionszeichen bzw. runde Klammern in Folge
([.,;:\?!\])	\1	Komma nach best. Satzzeichen: Entfernen!
([!])([.,;:^0130^0132^0145^0147\?!\])	\2	Ausrufezeichen vor best. Satzzeichen o. linkem Anführungszeichen: Entfernen!
([\?])([.,;:^0130^0132^0145^0147\?!\])	\2	Fragezeichen vor best. Satzzeichen o. linkem Anführungszeichen: dito
([\[\](\^0130^0132^0145^0147])([]{1;})	\1	2 oder mehr ungeschützte Leerzeichen nach öffnenden typografischen Anführungszeichen
([\^0130^0132^0145^0147])([])([0-9A-z])	\1\3	Alphanummerischer Text mit ungeschütztem Leerzeichen nach linken Anführungszeichen: Leerzeichen entfernen!
([\^0145^0146^0147^0148])![s]	\1^32	Leerzeichen fehlt hinter rechtem typografischem Anführungszeichen (halb o. doppelt): Einfügen!
([\^s]){2;}	\1	2 oder mehr geschützte Leerzeichen in Folge: Eins erhalten!
([0-9])-([0-9])	\1^=\2	Bindestrich zwischen zwei Zahlen durch Halbgeviertstrich ersetzen!
([0-9A-z])([])([\^0145^0146^0147^0148])	\1\3	Alphanummerischer Text mit ungeschütztem Leerzeichen vor schließendem (d. h. rechten) Anführungszeichen: Leerzeichen entfernen!
([\?])([\^0130^0132^0145^0147])	\2	Fragezeichen direkt vor öffnenden (d. h. linken) Anführungszeichen!
([\^+])([\^0145^0146^0147^0148])	\2	Geviertstich direkt vor schließendem (d. h. rechtem) typografischen Anführungszeichen: Geviertstrich entfernen!
([\^=])([\^0145^0146^0147^0148])	\2	Halbgeviertstrich direkt vor schließendem (d. h. rechtem) typografischen Anführungszeichen: Halbgeviertstrich entfernen!
([]{1;})([.,;:\?!\^0145^0147\)\])	\2	Leerzeichen vor Satzzeichen, schließenden typograf. Anführungszeichen, runden, geschweiften bzw. eckigen Klammern
([\^0145])([.,;:\^+])	\1	Komma, Semikolon, Doppelpunkt o. Geviertstrich nach halbem öffnendem (d. h. rechtem) Anführungszeichen: Satzzeichen löschen!
([\^0146])([.,;:\^+])	\1	Komma, Semikolon, Doppelpunkt o. Geviertstrich nach halbem öffnendem (d. h. rechtem) Anführungszeichen: Satzzeichen löschen!
[]{1;}-	-	Bindestrich mit 1 oder mehr führenden Leerzeichen: Diese entfernen!
-[]{1;}	-	Bindestrich mit 1 oder mehr nachfolgendem Leerzeichen: Diese entfernen!
[]{2;}	^32	2 oder mehr Leerzeichen in Folge: Überflüssige entfernen!
[-]{2}	^=	2 Bindestriche in Folge: Umwandeln in Halbgeviertstrich
[-]{3}	^+	3 Bindestriche in Folge: Umwandeln in Geviertstrich
[^t]{2;}	^9	2 oder mehr Tabulatorzeichen in Folge: Einen erhalten!
[]{1; }^13	^p	1 oder mehr Leerzeichen vor Absatzmarke: Leerzeichen entfernen!
^[]{1;}	^+	Geviertstrich mit 1 oder mehr nachfolgenden Leerzeichen: Diese entfernen!

<code>^=[]{1;}</code>	<code>^=</code>	Halbgeviertstrich mit 1 oder mehr nachfolgenden Leerzeichen: Dito!
<code>([!])^0133</code>	<code>^32^0133</code>	Fehlendes Leerzeichen vor horizontaler Ellipse: Einfügen!
<code>^0133([!])</code>	<code>^0133^32 \1</code>	Fehlendes Leerzeichen hinter horizontaler Ellipse: Einfügen!
<code>^13{2;}</code>	<code>^p</code>	2 oder mehr Absatzmarken in Folge: Eine erhalten!
<code>^34{2;}</code>	<code>^34</code>	2 oder mehr gerade doppelte Anführungszeichen in Folge: Eins erhalten!
<code>^39{2;}</code>	<code>^39</code>	2 oder mehr gerade halbe Anführungszeichen in Folge: Eins erhalten!
<code>^1{1;}</code>	<code>^p</code>	1 oder mehr manuelle Seitenumbrüche: Absatzmarke einfügen!
<code>^13[]{1;}</code>	<code>^p</code>	1 oder mehr Leerzeichen nach Absatzmarke: Diese erhalten! Siehe: <code>^p[^w]{1;}</code>
<code>^t{1;}</code>	<code>^t</code>	1 oder mehr manuelle Zeilenumbrüche: Nur einen erhalten!
<code>([0-9A-z])(^0130)</code>	<code>\1^32\2</code>	Alphanumerisches Zeichen direkt vor deutschem öffnendem typografischem halbem Anführungszeichen: Leerzeichen einfügen!
<code>(.)(^0130)</code>	<code>\1^32\2</code>	Punkt, dito
<code>(,)(^0130)</code>	<code>\1^32\2</code>	Komma, dito
<code>([!])(^0130)</code>	<code>\1^32\2</code>	Ausrufzeichen, dito
<code>([\?])(^0130)</code>	<code>\1^32\2</code>	Fragezeichen, dito
<code>([!])(^0130)</code>	<code>\1^32\2</code>	Fehlendes Leerzeichen vor deutschen öffnenden typografischen halben Anführungszeichen: Leerzeichen einfügen!
<code>(,)([.:;^=\+\?\!])</code>	<code>\2</code>	Komma vor Punkt, Semikolon, Doppelpunkt, Halbgeviertstrich, Geviertstrich, Fragezeichen, Ausrufezeichen: Komma entfernen!
<code>([.:;^=\+\?\!])(,)</code>	<code>\1</code>	Komma nach Punkt, Semikolon, Doppelpunkt, Halbgeviertstrich, Geviertstrich, Fragezeichen, Ausrufezeichen: Komma entfernen!
<code>([]{1;})([.,;:\?\!^0145^0147\)\}\])</code>	<code>\2</code>	Leerzeichen vor Satzzeichen, schließenden typograf. Anführungszeichen, runden, geschweiften bzw. eckigen Klammern entfernen
<code>([.,;:\?\!^0145^0147\)\}\])([]{2;})</code>	<code>\1^32</code>	Leerzeichen nach Satzzeichen, schließenden typograf. Anführungszeichen, runden, geschweiften bzw. eckigen Klammern entfernen
<code>([A-z])(,)([A-z])</code>	<code>\1\2^32\3</code>	Fehlendes Leerzeichen hinter Komma zwischen Satzteilen
<code>([.\?\!])([A-Z])</code>	<code>\1^32\2</code>	'Fehlendes Leerzeichen hinter dem Ende (.?!) eines Satzes einfügen
<code>([0-9])0([0-9])</code>	<code>\10\2</code>	Großbuchstabe '0' innerhalb einer Zahl durch Null (0) ersetzen
<code>0([0-9])</code>	<code>0\1</code>	Großbuchstabe '0' vor einer Zahl durch Null (0) ersetzen
<code>([0-9])0</code>	<code>\10</code>	Großbuchstabe '0' hinter einer Zahl durch Null (0) ersetzen
<code>(^32)(-)(^32)</code>	<code>\2</code>	Leerzeichen vor und nach Bindestrich entfernen
<code>[^w]{1;}^p</code>	<code>^p</code>	1 oder mehr Leerzeichen vor Absatzmarke: Leerzeichen entfernen!
<code>^p[^w]{1;}</code>	<code>^p</code>	1 oder mehr Leerzeichen nach Absatzmarke: Leerzeichen entfernen!

Tabelle 1: Suchbegriffe und Ersetzungen

3 Zieldokument

Das Zieldokument wird mit einem Benutzerdialog ausgewählt. Das Zieldokument ist das Objekt für das Finden und Ersetzen mit einer externen Tabelle, die sich im Referenzdokument (auch Quelldokument genannt) befindet.

4 Prozeduren

Das Zusammenspiel zwischen Referenz- und Zieldokument erfolgt durch VBA-Prozeduren und Funktionen, die sich in den Modulen mit dem Namen "ReplaceFromTable" und "FindFromTable" des Dokuments "Replacer.docm" befinden (s. Abbildung 1)

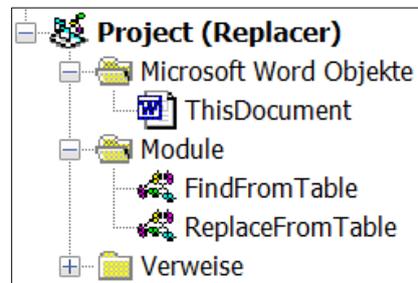


Abbildung 1: Projekt "Replacer"

Der VBA-Code in den genannten Modulen ist ähnlich. Sie verwenden aber unterschiedliche Tabellen des Referenzdokuments und unterscheiden sich im Umfang der jeweiligen Aufgabe.

Nur die VBA-Prozeduren im Modul "ReplaceFromTable" werden im Folgenden präsentiert:

5 Prozeduren und Funktionen im Modul "ReplaceFromTable"

Das Modul "Replace fromTable" umfasst folgende 7 Objekte (s. Tabelle 2):

Name	Typ	Aufgabe
ReplaceFromTable	sub	Finden und Ersetzen mit Tabelle im Referenzdokument und Benutzerdialog.
BuildDynamicArrays	sub	Zwei dynamische Arrays erzeugen für das Finden u. Ersetzen mit Einträgen der Tabelle im Referenzdokument
OpenWordFile	function	Word-Datei auswählen und öffnen
GetCellContent	function	Inhalt einer Tabellenzelle zurückgeben
FindWhiteSpace	function	Vorkommen des Zeichens (^w) für 'White Space' auswerten
FindReplaceCount	function	Benutzerdialog für Suchen und Ersetzen mit Rückgabe der Zahl der Ersetzungen
ExpandedRange	function	Fundstelle auf ganze Textzeile ausweiten

Tabelle 2: Objekte der Anwendung

Die Hierarchie der Objekte ist aus Abbildung 2 ersichtlich:

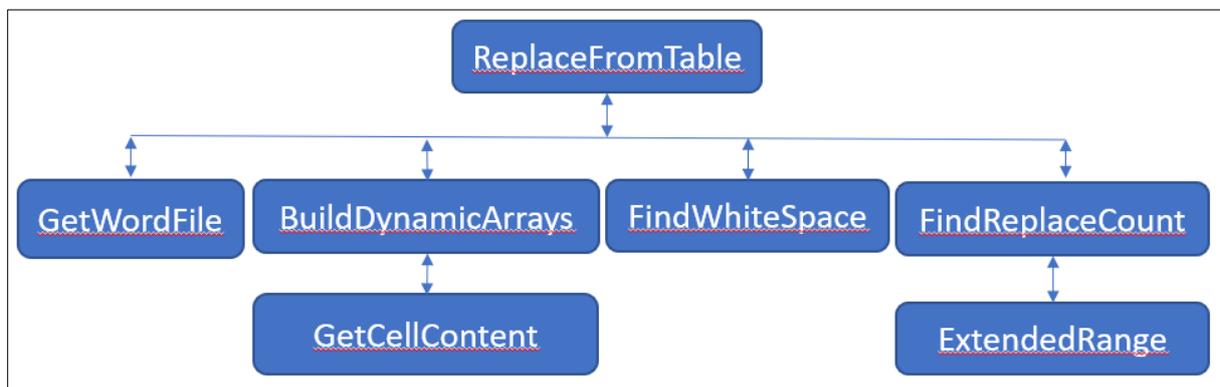


Abbildung 2: Struktogramm der Anwendung "ReplaceFromTable"

6 VBA-Quellcode der Anwendung

```

1 Option Explicit
2 Option Base 1
3 Public arrFind() As String ' Dyn. Array f. Suchen
4 Public arrRepl() As String ' Dyn. Array f. Ersetzen
5
6 Private Sub ReplaceFromTable()
7 ' Aufgabe: Finden und Ersetzen mit Tabelle im Referenzdokument
8 ' und Benutzerdialog
9 Dim bolSwitch As Boolean ' Schalter für 'MatchWildcards'
10 Dim docTgt As Document ' Zieldokument
11 Dim docRef As Document ' Referenzdokument mit Tabelle
12 Dim lngCount As Long ' Zahl der Ersetzungen
13 Dim lngIdx As Long ' Zeilenindex d. Tabelle
14 Dim lngRows As Long ' Zeilenzahl der Tabelle
15 Dim lngLoop As Long ' Schleifen- bzw. Zeilenzähler
16 Dim lngColorIdx As Long ' Farbe z. Hervorhebung d. Fundstelle(n)
17 Dim objTbl As Table ' Tabelle im Referenzdokument
18 Dim objRng As Range ' Textbereich im Zieldokument
19 Dim strFullNm As String ' Vollst. Dateiname der Zieldatei
20 Dim strFind As String ' Suchen nach ... (Suchbegriff)
21 Dim strRepl As String ' Ersetzen durch ... (Ersetzung)
22
23 On Error GoTo Error_Point ' Fehlerbehandlung
24
25 ' Zieldokument interaktiv auswählen
26 strFullNm = OpenWordFile()
27 If Len(strFullNm) > 0 Then
28     Set docTgt = Documents.Open(strFullNm, ReadOnly:=False, _
29         AddToRecentFiles:=False, Visible:=False)
30 Else
31     MsgBox "Abbruch: Auswahl d. Zieldokuments ist fehlgeschlagen!", _
32         vbExclamation, "ReplaceFromTable"
33     Exit Sub
34 End If
35
36 ' Bildschirmaktualisierung deaktivieren
37 Application.ScreenUpdating = False
38
39 ' Möglicherweise schon vorhandene Hervorhebungen
40 ' im Zieldokument entfernen
41 docTgt.Range.HighlightColorIndex = wdNoHighlight
42
43 ' Farbe zum Hervorheben bestimmen
44 lngColorIdx = Options.DefaultHighlightColorIndex
45 Options.DefaultHighlightColorIndex = wdYellow
46
47 ' Zwei dynamische Arrays für Finden u. Ersetzen erzeugen
48 ' mit Tabelle im Referenzdokument
49 Call BuildDynamicArrays
50
51 ' Gesante Zeilenzahl d. dynamischen Arrays bestimmen
52 lngRows = UBound(arrFind)
53
54 ' Zieldokument auswählen

```

```

55 docTgt.Select
56
57 ' Zählschleife ausführen
58 For lngIdx = 1 To lngRows
59     strFind = arrFind(lngIdx)
60     strRepl = arrRepl(lngIdx)
61     ' Schalter (True/False) setzen f. Parameter 'MatchWildcards' d. Suche
62     bolSwitch = FindWhiteSpace(strFind)
63     If Len(strFind) > 0 And Len(strRepl) > 0 Then
64         ' Nur Haupttext u. Fußnoten des Zieldokuments durchlaufen
65         For Each objRng In docTgt.StoryRanges
66             If objRng.StoryType = wdMainTextStory Or _
67                objRng.StoryType = wdFootnotesStory Then
68                 ' Zahl der Ersetzungen erhöhen
69                 lngCount = lngCount + FindReplaceCount(docTgt, _
70                    strFind, strRepl, bolSwitch)
71             End If
72         Next objRng
73     End If
74 Next lngIdx
75 MsgBox "Normales Ende: " & CStr(lngCount) & _
76     " Ersetzung(en) wurde(n) vorgenommen im Zieldokument '" _
77     & docTgt.Name & "'", vbExclamation, "ReplaceFromTable"
78 Exit_Point:
79     ' Bildschirmaktualisierung deaktivieren
80     Application.ScreenUpdating = False
81     ' Farbe zum Hervorheben zurücksetzen
82     Options.DefaultHighlightColorIndex = lngColorIdx
83     ' Zieldokument speichern
84     With docTgt
85         .Range(0, 0).Select
86         If Not .Saved Then .Save
87         .Close SaveChanges:=wdSaveChanges,
88 OriginalFormat:=wdOriginalDocumentFormat
89     End With
90     With Documents
91         If .Count > 0 Then
92             .Close SaveChanges:=wdDoNotSaveChanges,
93 OriginalFormat:=wdOriginalDocumentFormat
94         End If
95     End With
96     Application.Quit
97     Exit Sub
98 Error_Point:
99     MsgBox "Laufzeitfehler " & lngIdx & " " & Err.Number & ", " &
100 Err.Description, _
101         vbInformation, ActiveDocument.Name
102     Resume Exit_Point
103 End Sub
104
105 Private Function FindReplaceCount( _
106     ByRef docTgt As Document, _
107     ByVal strFind As String, _
108     ByVal strRepl As String, _
109     ByVal bolTrueFalse) As Long

```

```

110 ' Aufgabe: Benutzerdialog f. Suchen u. Ersetzen
111 ' mit Rückgabe d. Zahl d. Ersetzungen
112 Dim objRng As Range ' Textbereich
113 Dim intLineLen As Integer ' Zeilenlänge der Fundstelle
114 Dim lngCount As Long ' Zähler f. Ersetzungen
115 Dim lngRetVal As Long ' Rückgabewert der Funktion
116 Dim strLine As String ' Text in Zeile der Fundstelle
117 docTgt.Select
118 Set objRng = docTgt.Content
119 ' Anfangswert für Zähler der Ersetzungen setzen
120 FindReplaceCount = 0
121 With objRng.Find
122     .ClearFormatting
123     .Text = strFind
124     With .Replacement
125         .ClearFormatting
126         .Text = strRepl
127         .Highlight = True
128     End With
129     .Forward = True
130     .Wrap = wdFindStop
131     .Format = False
132     .MatchCase = False
133     .MatchWholeWord = False
134     .MatchAllWordForms = False
135     .MatchSoundsLike = False
136     .MatchWildcards = bolTrueFalse
137     .Execute
138     Do While .Found = True
139         ' Fundstelle ausweiten f. Meldung im Dialogfeld
140         strLine = ExpandedRange(objRng)
141         strLine = strLine & Space(1) & Chr(133) & vbCrLf & vbCrLf
142         Select Case MsgBox(Prompt:=strLine & "Ersetzung durchführen?",
143 Buttons:=vbYesNoCancel, _
144 Title:="Finden: " & strFind & " " & Chr(151) & " Ersetzen:
145 " & strRepl)
146     Case vbYes
147         .Execute Replace:=wdReplaceOne
148         objRng.HighlightColorIndex = wdYellow
149         ' Zähler f. Ersetzungen erhöhen
150         lngCount = lngCount + 1
151     Case vbCancel
152         Exit Do ' Scheife beenden
153     Case Else: ' nichts tun
154     End Select
155     objRng.Collapse Direction:=wdCollapseEnd
156     .Execute
157     Loop
158 End With
159 ' Zähler d. Ersetzungen zurückgeben
160 FindReplaceCount = lngCount
161 End Function

162 Sub BuildDynamicArrays()

```

```

163 ' Zwei dynamische Arrays für Finden u. Ersetzen erzeugen
164 ' mit Tabelle im Referenzdokument
165 Dim docRef      As Document ' Referenzdokument
166 Dim objTbl     As Table    ' Tabelle im Referenzdokument
167 Dim lngIdx     As Long     ' Zeilenindex d. Tabelle
168 Dim lngRows    As Long     ' Zeilenzahl d. Tabelle
169 Dim lngLoop    As Long     ' Schleifenzähler
170 ' Referenzdokument bestimmen
171 Set docRef = Documents.Open("C:\Users\" & Environ("Username") & _
172     "\Documents\CheckDocuments.docm", ReadOnly:=True, _
173     AddToRecentFiles:=False, Visible:=False)
174 ' Tabelle im Referenzdokument bestimmen f. das Finden u. Ersetzen
175 Set objTbl = docRef.Tables(3)
176 ' Dynamische Arrays dimensionieren
177 lngRows = objTbl.Rows.Count
178 ReDim arrFind(lngRows - 1)
179 ReDim arrRepl(lngRows - 1)
180 ' Dyn. Arrays f. Suchen/Ersetzen populieren
181 lngIdx = 1
182 For lngLoop = 2 To lngRows
183     arrFind(lngIdx) = GetCellContent(objTbl.Cell(lngLoop, 1))
184     arrRepl(lngIdx) = GetCellContent(objTbl.Cell(lngLoop, 2))
185     lngIdx = lngIdx + 1
186 Next lngLoop
187 ' Schreibgeschütztes Referenzdokument schließen
188 docRef.Close SaveChanges:=wdDoNotSaveChanges
189 Set docRef = Nothing
190 End Sub

191 Function OpenWordFile() As String
192     ' Aufgabe: Word-Datei auswählen und öffnen
193     Dim strFullNm As String    ' vollständiger Dateiname d. Zieldokuments
194     Dim lngRtnCode As Long    ' Rückgabewert von MsgBox
195     On Error GoTo Err_Point   ' Fehlerroutine
196     OpenWordFile = vbNullString
197     With Application.FileDialog(msoFileDialogOpen)
198         .Title = "Dateiauswahl"
199         .ButtonName = "Auswählen"
200         .AllowMultiSelect = False
201         .InitialFileName = ThisDocument.Path
202         With .Filters
203             .Clear
204             .Add "Word-Dateien (*.docx)", "*.docx", 1
205             .Add "Word-Dateien (*.docm)", "*.docm", 2
206         End With
207         .FilterIndex = 1
208         If .Show = True Then
209             strFullNm = CStr(.SelectedItems(1)) ' Pfadname + Dateiname
210             If Len(strFullNm) > 0 And Dir(strFullNm) <> vbNullString Then
211                 OpenWordFile = strFullNm
212             Else
213                 MsgBox "Zieldokument kann nicht geöffnet werden", _
214                     vbExclamation, "OpenWordFile"
215             End If
216         Else

```

```

217         MsgBox "Es wurde nichts ausgewählt!", vbOKOnly, "Dateiauswahl"
218     End If
219 End With
220 Exit_Point:
221     Exit Function
222 Err_Point:
223     MsgBox "Laufzeitfehler: " & Err.Description, vbCritical, "OpenWordFile"
224     Resume Exit_Point
225 End Function

226 Private Function FindWhiteSpace(ByVal strFind As String) As Boolean
227     ' Aufgabe: Vorkommen des Zeichens (^w) auswerten
228     ' ^w steht für Leerraum/Leerzeichen/Tabulator/etc.
229     If InStr(strFind, "^w") > 0 Then
230         FindWhiteSpace = False
231     Else
232         FindWhiteSpace = True
233     End If
234 End Function

235 Public Function GetCellContent(CEL As Word.Cell) As String
236     ' Aufgabe: Gibt den Inhalt einer Zelle d. Tabelle zurück.
237     GetCellContent = Trim(Left(CEL.Range.Text, Len(CEL.Range.Text) - 2))
238 End Function

239 Function ExpandedRange(objRng As Range) As Range
240     ' Aufgabe: Fundstelle auf ganze Textzeile ausweiten
241     Dim intLineLen As Integer ' Zeilenlänge d. Fundstelle
242     ExpandedRange = vbNullString
243     ' Fundstelle auswählen
244     objRng.Select
245     ' Fundstelle aufbereiten f. Meldung im Dialogfeld
246     With Selection
247         intLineLen = .Expand(wdLine) + 1 ' Zeilenlänge d. Fundstelle
248         .HomeKey Unit:=wdLine ' Zum Anfang d. Zeile springen
249         ' Auswahl nach rechts ausweiten
250         .MoveRight Unit:=wdCharacter, Count:=intLineLen, Extend:=wdExtend
251         ' Expandierten Textbereich zurückgeben
252         ExpandedRange = .Text
253     End With
254 End Function

```