

Überflüssige Leerzeichen entfernen

Remove Superfluous Spaces

Autor: Dr. V. Thormählen

Inhalt

Inhalt	1
Abbildungen	1
Listings	1
Tabellen.....	1
Aufgabe	2
Anwendungsbeispiel	2

Abbildungen

Abbildung 1: Text mit überflüssigen Leerzeichen vor bzw. nach einem Komma.....	3
Abbildung 2: Text ohne überflüssige Leerzeichen vor bzw. nach einem Komma	3

Listings

Listing 1: Finden und Ersetzen im Haupttext eines Zieldokuments	6
Listing 2: Finden und Ersetzen in ausgewählten Bereichen des Zieldokuments.....	7
Listing 3: Funktionsprozedur zum Finden, Ersetzen und Zählen im Zieldokument.....	8

Tabellen

Tabelle 1: Einträge im Quelldokument für das Finden und Ersetzen überflüssiger Leerzeichen	2
Tabelle 2: Tabelleneinträge für das Anwendungsbeispiel	2
Tabelle 3: WdStoryType-Aufzählung (Quelle: Microsoft Word)	4

Aufgabe

In diesem Beitrag wird beschrieben, wie überflüssige Leerzeichen aus einem WORD-Zieldokument mit einer VBA-Prozedur automatisch entfernt werden können. Dabei wird Tabelle 1 benutzt, deren Einträge im aktuell sichtbaren WORD-Quelldokument enthalten sind:

Zeile	Finden	Ersetzen	Aufgabe
2	([]{1;})(^13)	\2	Führende Leerzeichen aus Absatz entfernen.
3	(^13)([]{1;})	\1	Nachgest. Leerzeichen aus Absatz entfernen.
4	() {2;}	\1	Zwei oder mehr Leerzeichen durch 1 LZ ersetzen.
5	([]{1;})([.,;:\?!\\"\']\W)	\2	Mehrere Leerzeichen vor Satzzeichen entfernen.
6	([.,;:\?!\\"'\W])([]{1;})	\1\2	Mehrere Leerzeichen nach Satzzeichen entfernen.
7	([]{2;})([!])	\2	Unnötige Leerzeichen zwischen Wörtern entfernen.
8	[^13^]{2;}	^p	Zeilen-/Absatzumbruch auf 1 je „echten“ Umbruch begrenzen.
9	([]{1;})([]{1;})	^p	Leerzeichen und Tabulatoren vor Absatzumbrüchen entfernen.
10	[]{2;}	^s	Zwei oder mehr (geschützte) Leerzeichen entfernen.
11	[^t]{2;}	^t	Zwei oder mehr Tabulatoren entfernen.
12	[]{1;}{1;}	^p	Leerzeichen/Tabulatoren vor Absatzumbrüchen entfernen.
13	([^02^0146^0148])([]{2;})	\1	Überschüssige Leerzeichen nach Tabulatoren oder schließenden Ausrufezeichen (einfach oder doppelt) entfernen.
14	([]{2;})([^02^0145^0147])	\2	Überschüssige Leerzeichen vor Tabulatoren oder öffnenden (einfachen/doppelten) Ausrufezeichen entfernen.
15	([]{1;})(^2)	\2	Alle Leerzeichen vor Fußnotennummer entfernen.
16	([]{1;})([.!\"])	\2	Alle Leerzeichen vor Satzendezeichen entfernen.
17	([]{1;})([()\[\]])	\2	Alle Leerzeichen vor schließender (runde, eckige, geschweifte) Klammer entfernen.
18	([()\[\]])([]{1;})	\1	Alle Leerzeichen nach öffnender (runde, eckige, geschweifte) Klammer entfernen.

Tabelle 1: Einträge im Quelldokument für das Finden und Ersetzen überflüssiger Leerzeichen

Das WORD-Zieldokument (docTgt genannt) wird vom Benutzer im Quellcode der Subprozedur *ReplaceFromTable* (s. Listing 1) mit den Codezeilen 20 ff bestimmt und ist ggf. anzupassen.

Anwendungsbeispiel

Als Beispiel werden der 5. und 6. Eintrag in Tabelle 1 betrachtet, das heißt:

Zeile	Finden	Ersetzen	Aufgabe
5	([]{1;})([.,;:\?!\\"'\W])	\2	Mehrere Leerzeichen vor Satzzeichen entfernen.
6	([.,;:\?!\\"'\W])([]{1;})	\1\2	Mehrere Leerzeichen nach Satzzeichen entfernen.

Tabelle 2: Tabelleneinträge für das Anwendungsbeispiel

Abbildung 1 beinhaltet einen Text, der vor bzw. nach einem Komma multiple Leerzeichen enthält. Die entsprechenden Textstellen sind gelb markiert.

Die überflüssigen Leerzeichen werden mit einer VBA-Prozedur automatisch entfernt (s. Listing 1).

Der Inhalt in Abbildung 2 gleicht dem in Abbildung 1, aber ohne multiple Leerzeichen vor bzw. nach dem Komma. Die überflüssigen Leerzeichen wurden entfernt.

Video provides a powerful way to help you prove your point. When you click Online Video , you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar. Click Insert and then choose the elements you want from the different galleries.

Themes and styles also help keep your document coordinated. When you click Design and choose a new Theme , the pictures, charts, and SmartArt graphics change to match your new theme. When you apply styles , your headings change to match the new theme.

Abbildung 1: Text mit überflüssigen Leerzeichen vor bzw. nach einem Komma

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar. Click Insert and then choose the elements you want from the different galleries.

Themes and styles also help keep your document coordinated. When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme. When you apply styles , your headings change to match the new theme.

Abbildung 2: Text ohne überflüssige Leerzeichen vor bzw. nach einem Komma

Bereiche eines WORD-Dokuments		
Name	Wert	Beschreibung
wdMainTextStory	1	Haupttext
wdFootnotesStory	2	Fußnoten
wdEndnotesStory	3	Endnoten
wdCommentsStory	4	Kommentar
wdTextFrameStory	5	Textrahmen
wdEvenPagesHeaderStory	6	Gerade Kopfzeile
wdPrimaryHeaderStory	7	Primäre Kopfzeile
wdEvenPagesFooterStory	8	Gerade Fußzeile
wdPrimaryFooterStory	9	Primäre Fußzeile
wdFirstPageHeaderStory	10	Erste Kopfzeile
wdFirstPageFooterStory	11	Erste Fußzeile
wdFootnoteSeparatorStory	12	Fußnotentrennlinie
wdFootnoteContinuationSeparatorStory	13	Fußnotenfortsetzungstrennlinie
wdFootnoteContinuationNoticeStory	14	Fußnotenfortsetzungshinweis
wdEndnoteSeparatorStory	15	Endnotentrennlinie
wdEndnoteContinuationSeparatorStory	16	Endnotenfortsetzungstrennlinie
wdEndnoteContinuationNoticeStory	17	Endnotenfortsetzungshinweis

Tabelle 3: WdStoryType-Aufzählung (Quelle: Microsoft Word)

```

1 Sub ReplaceFromTable()
2   Dim docSrc As Document ' Quelldokument (enthält d. Ersetzungstabelle)
3   Dim docTgt As Document ' Zieldokument
4   Dim lngHiLight As Long ' Aktuelle Hervorhebungsfarbe
5   Dim lngRow As Long ' Zeilenzähler f. die Tabelle im Quelldokument
6   Dim rngFnd As Range ' Suchbegriff
7   Dim rngRpl As Range ' Ersatzbegriff
8   Dim strTgtNm As String ' Dateipfad zum Zieldokument
9   Dim strSrcNm As String ' Dateipfad zum Quelldokument
10
11   ' Aktuelle Hervorhebungsfarbe speichern, damit sie später zurückgesetzt werden kann.
12   lngHiLight = Options.DefaultHighlightColorIndex
13   Options.DefaultHighlightColorIndex = wdYellow
14
15   On Error GoTo Err_Handler
16   Application.ScreenUpdating = False
17
18   strTgtNm = Environ("USERPROFILE") & Application.PathSeparator & "Documents\docTarget.docx"
19   If Dir(strTgtNm) > vbNullString Then
20     ' docTgt: WORD-Zieldokument, in dem der Ersatz erfolgen soll.
21     Set docTgt = Documents.Open(Filename:=strTgtNm, ReadOnly:=False, AddToRecentFiles:=True)
22   Else
23     MsgBox "Abbruch: Der Pfad zum Zieldokument ist ungültig", vbCritical
24     Exit Sub
25   End If
26
27   strSrcNm = Environ("USERPROFILE") & Application.PathSeparator &
28   "Documents\Remove_Superfluous_Spaces.docm"
29   If Dir(strSrcNm) > vbNullString Then
30     ' docSrc: WORD-Zieldokument, in dem der Ersatz erfolgen soll.
31     Set docSrc = Documents.Open(Filename:=strSrcNm, ReadOnly:=True)
32   Else
33     MsgBox "Abbruch: Der Pfad zum Quelldokument ist ungültig", vbCritical
34     Exit Sub
35   End If
36
37   ' docSrc: WORD-Quelldokument, in dem sich die Ersetzungstabelle befindet.
38   docSrc.Select
39   With docSrc.Tables(1)
40     For lngRow = 2 To .Rows.Count
41       Set rngFnd = .Cell(lngRow, 2).Range
42       rngFnd.End = rngFnd.End - 1
43       Set rngRpl = .Cell(lngRow, 3).Range
44       rngRpl.End = rngRpl.End - 1
45       docTgt.Activate
46       With Selection.Find
47         .Forward = True
48         .Wrap = wdFindContinue
49         .MatchWholeWord = True
50         .MatchWildcards = True
51         .MatchSoundsLike = False
52         .MatchAllWordForms = False
53         .Format = True
54         .MatchCase = False
55         .Text = rngFnd
56       With .Replacement
57         .Highlight = True
58         .Text = rngRpl

```

```

59         End With
60         .Execute Replace:=wdReplaceAll
61     End With
62     Next lngRow
63 End With
64 MsgBox "Normales Ende"
65 Exit_Point:
66     Options.DefaultHighlightColorIndex = lngHiLight
67     docSrc.Close SaveChanges:=wdDoNotSaveChanges
68     Application.ScreenUpdating = True
69     docTgt.Activate
70     Exit Sub
71 Err_Handler:
72     MsgBox "Laufzeitfehler: " & Err.Number & ", " & Err.Description & ", " & Err.Source
73     Resume Exit_Point
74 End Sub

```

Listing 1: Finden und Ersetzen im Haupttext eines Zieldokuments

```

1  Public Sub FindReplaceAnywhere
2  Dim docTgt      As Document      ' Zieldokument
3  Dim docSrc      As Document      ' Quelldokument
4  Dim lngFound    As Long          ' Zahl der Ersetzungen im Zieldokument
5  Dim lngJunk     As Long          ' Testergebnis wg. Müll
6  Dim rngStory    As Range         ' Bereich des Zieldokuments
7  Dim strFnd      As String        ' finden (Suchbegriff)
8  Dim strRpl      As String        ' ersetzen durch (Ersatzbegriff)
9  Dim strPath     As String        ' Pfad zum Zieldokument
10 Application.ScreenUpdating = False
11 ' Öffnen des Zieldokuments.
12 Set docTgt = Documents.Open(FileName:=Environ("UserProfile") & "\Documents\docTarget.docx")
13 ' MsgBox docTgt.FullName
14 ' Öffnen des Quelldokuments.
15 Set docSrc = Documents.Open(Environ("UserProfile") & "\Documents\Remove_Superfluous_Spaces.docm")
16 ' MsgBox docSrc.FullName
17 ' Das Problem der übersprungenen leeren Kopf-/Fußzeile beheben.
18 lngJunk = docTgt.Sections(1).Headers(1).Range.StoryType
19 ' Suchbegriff: Beliebige Leerzeichen vor und nach dem Komma.
20 strFnd = "([ ]{1;})([,])([ ]{1;})"
21 ' Ersatzbegriff: Keine Leerzeichen vor dem Komma, aber 1 danach.
22 strRpl = "\2 "
23 ' Anfangswert f. Zahl d. Ersetzungen.
24 lngFound = 0
25 ' Im Zieldokument alle Bereiche durchlaufen.
26 For Each rngStory In docTgt.StoryRanges
27     Do
28         lngFound = lngFound + fctFRInStory(rngStory, strFnd, strRpl, True)
29         On Error Resume Next
30         Select Case rngStory.StoryType
31             ' Nur bestimmte verknüpfte Bereiche durchlaufen.
32             Case 6, 7, 8, 9, 10, 11
33                 lngFound = lngFound + fctFRInStory(rngStory, strFnd, strRpl, True)
34             Case Else
35                 ' Nichts tun!
36             End Select
37         On Error GoTo 0
38         ' Nächsten verlinkten Bereich holen, falls vorhanden.
39         Set rngStory = rngStory.NextStoryRange
40     Loop Until rngStory Is Nothing
41 Next rngStory
42 MsgBox "Fertig! " & Str(lngFound) & " Ersetzung(en) im Zieldokument vorgenommen.", vbInformation
43 End Sub

```

Listing 2: Finden und Ersetzen in ausgewählten Bereichen des Zieldokuments

```

1  Function fctFRInStory(ByVal rngStory As Range, _
2      ByVal strFnd As String, _
3      ByVal strRpl As String, _
4      ByVal bolWildCards As Boolean) As Long
5      With rngStory.Find
6          .ClearFormatting
7          .Replacement.ClearFormatting
8          Do While .Execute(FindText:=strFnd, _
9              MatchWildcards:=bolWildCards, _
10                 Forward:=True, _
11                 Wrap:=wdFindContinue, _
12                 ReplaceWith:=strRpl, _
13                 Replace:=wdReplaceOne) = True
14              rngStory.Collapse Direction:=wdCollapseEnd
15              fctFRInStory = fctFRInStory + 1
16          Loop
17      End With
18  End Function

```

Listing 3: Funktionsprozedur zum Finden, Ersetzen und Zählen im Zieldokument