

Suchen und Ersetzen in Dokumentkomponenten

Volker Thormählen, 24. Mai 2017

1 Aufgabe

In diesem Beitrag wird beschrieben, wie bestimmte Komponenten eines WORD-Dokuments durchlaufen werden können, um dort mit der in WORD standardmäßig eingebauten Funktionalität „Suchen und Ersetzen“ die erforderliche(n) Korrektur(en) interaktiv durchzuführen.

In einem Word-Dokument können bis zu 17 Komponenten vorkommen. Dazu zählt nicht nur der sog. *Haupttext*, sondern möglicherweise auch andere Dokumentkomponenten wie Fußnoten, Endnoten, Kopfzeilen, Fußzeilen usw. (vgl. Listing 3)

2 Lösung

Die Lösung beinhaltet 2 Subprozeduren und 1 Funktionsprozedur:

- Die Subprozedur `FindReplaceAnywhere` entfernt 1 Leerzeichen vor Satzzeichen bzw. schließenden Klammern im aktuellen Dokument (s. Listing 1)
- Die Subprozedur `SearchAndReplaceInStory` vollzieht die gerade genannte Aufgabe für die jeweilige Dokumentkomponente (s. Listing 2)
 - Codezeile 16 bewirkt die Markierung einer Fundstelle
 - Codezeile 19 ff beinhalten einen Benutzerdialog, wie bezüglich der Fundstelle entschieden und vorgegangen wird
- Die Funktionsprozedur `StoryTypeDescript` ermittelt die jeweilige Bezeichnung einer Dokumentkomponente (s. Listing 3)

Die Anwendung endet mit folgender Erfolgsmeldung, falls nur 1 Fundstelle ermittelt und ersetzt wurde:

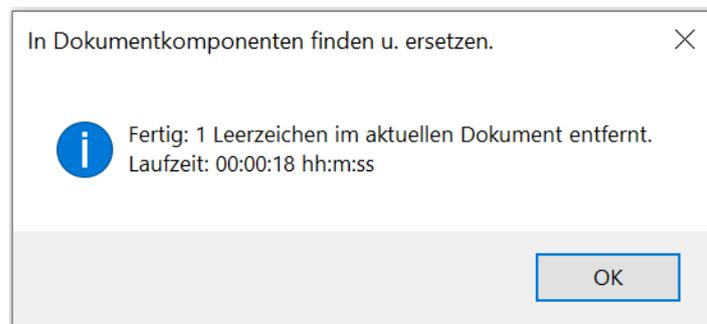


Abbildung 1: Erfolgsmeldung am Ende der Anwendung

3 VBA¹-Quellcode der Lösung

```
1 Sub FindReplaceAnywhere()  
2 ' Aufgabe: Leerzeichen entfernen vor Satzzeichen bzw. schließenden Klammern  
3 ' in best. Dokumentkomponenten.  
4 Const conTitle As String = "In Dokumentkomponenten finden u. ersetzen."  
5 Dim lngJunk As Long ' siehe weiter unten  
6 Dim lngChrCnt1 As Long ' Zeichenzahl im Dokument am Anfang  
7 Dim lngChrDiff As Long ' Entfernte Zeichenzahl  
8 Dim rngStory As Range ' Dokumentkomponente  
9 Dim dtmStart As Date ' Startzeit der aktuellen Subprozedur  
10 Dim strRunTm As String ' Laufzeit (in Stunden, Minuten, Sekunden)  
11 Dim arrFnd As Variant ' Datenbereich für 9 Suchbegriffe  
12 Dim arrRpl As Variant ' dito für zugehörige Ersetzungen  
13 On Error GoTo ErrorHandler  
14 dtmStart = Now  
15 lngChrCnt1 = ActiveDocument.Characters.Count  
16 arrFnd = Array(" ", " ;", " .", " :", " !", " ?", " )", " ]", " }")  
17 arrRpl = Array(" ", " ;", " .", " :", " !", " ?", " )", " ]", "}")  
18 ' Behebung d. übersprungenen leeren Header-Footer-Problems  
19 lngJunk = ActiveDocument.Sections(1).Headers(wdHeaderFooterPrimary).Range.StoryType  
20 ' Best. Dokumentkomponenten durchlaufen  
21 For Each rngStory In ActiveDocument.StoryRanges  
22 Do Until rngStory Is Nothing  
23 On Error Resume Next  
24 If Len(rngStory.Text) > 0 Then  
25 SearchAndReplaceInStory rngStory, arrFnd, arrRpl  
26 End If  
27 Select Case rngStory.StoryType  
28 Case 6, 7, 8, 9, 10, 11  
29 If Len(rngStory.Text) > 0 Then  
30 SearchAndReplaceInStory rngStory, arrFnd, arrRpl  
31 End If  
32 Case Else  
33 ' nichts tun!  
34 End Select  
35 ' Die nächste verknüpfte Dokumentkomponente ermitteln.  
36 On Error GoTo 0  
37 Set rngStory = rngStory.NextStoryRange  
38 Loop  
39 Next rngStory  
40 ' Laufzeit ermitteln  
41 strRunTm = Format(Now - dtmStart, "hh:mm:ss")  
42 ' Entfernte Zeichenzahl ermitteln  
43 lngChrDiff = lngChrCnt1 - ActiveDocument.Characters.Count  
44 MsgBox "Fertig: " & CStr(lngChrDiff) & " Leerzeichen im aktuellen Dokument entfernt." & _  
45 vbCrLf & "Laufzeit: " & strRunTm & " hh:m:ss", vbInformation, conTitle  
46 ExitPoint:  
47 On Error Resume Next  
48 Exit Sub  
49 ErrorHandler:  
50 MsgBox "Laufzeitfehler: " & Err.Number & ", " & Err.Description, vbCritical, conTitle  
51 Resume ExitPoint  
52 End Sub
```

Listing 1: VBA-Quellcode der Subprozedur 'FindReplaceAnywhere'

¹ Das Kürzel **VBA** steht für **Visual Basic for Applications**. Das ist eine proprietäre Skriptsprache der Firma Microsoft für die sog. Office-Suite.

```

1 Sub SearchAndReplaceInStory(ByVal rngStory As Range, _
2     ByVal arrFnd As Variant, _
3     ByVal arrRpl As Variant) _
4     Dim intChanges As Integer ' Zähler f. Änderungen
5     Dim intHits As Integer ' Trefferzähler
6     Dim intLoop As Integer ' Scheifenzähler
7     Dim lngRslt As Long ' Rückgabewert der MsgBox (Benutzerdialog)
8     With rngStory.Find
9         For intLoop = 0 To UBound(arrFnd)
10            .ClearFormatting
11            .Text = arrFnd(intLoop)
12            .Replacement.Text = ""
13            .Execute
14            Do While .Found
15                intHits = intHits + 1
16                rngStory.Duplicate.Select ' Aktuelle Fundstelle markieren.
17                ' Fensterinhalt verschieben, bis die Markierung darin angezeigt wird.
18                ActiveWindow.ScrollIntoView obj:=Selection.Range, Start:=True
19                lngRslt = MsgBox("Ersetzen:" & vbCr & vbTab & arrFnd(intLoop) & vbCr & _
20                    "durch:" & vbCr & vbTab & arrRpl(intLoop), vbYesNoCancel, _
21                    "In Dokumentkomponenten finden und ersetzen.")
22                Select Case lngRslt
23                    Case vbCancel
24                        Exit Sub
25                    Case vbYes
26                        rngStory.Text = arrRpl(intLoop)
27                        intChanges = intChanges + 1
28                    Case Else
29                        ' nichts tun!
30                End Select
31                rngStory.Collapse Direction:=wdCollapseEnd
32                .Execute
33            Loop
34            Next intLoop
35            If intHits > 0 Then
36                MsgBox CStr(intHits) & " Treffer in '" & StoryTypeDescript(rngStory.StoryType) & _
37                    "' mit " & CStr(intChanges) & " Änderung(en).", vbInformation, _
38                    "In Dokumentkomponenten finden und ersetzen."
39            End If
40        End With
41    End Sub

```

Listing 2: VBA-Quellcode der Subprozedur 'SearchAndReplaceInStory'

```

1 Function StoryTypeDescript(ByVal intStoryType As Integer) As String
2     ' Aufgabe: Bezeichnung der jeweiligen Dokumentkomponente zurückgeben.2
3     Dim varStory(17) As Variant
4     varStory(1) = "Haupttext"
5     varStory(2) = "Fußnote"
6     varStory(3) = "Endnote"
7     varStory(4) = "Kommentar"
8     varStory(5) = "Textrahmen"
9     varStory(6) = "Gerade Kopfzeile"
10    varStory(7) = "Primäre Kopfzeile"
11    varStory(8) = "Gerade Fußzeile"
12    varStory(9) = "Primäre Fußzeile"
13    varStory(10) = "Erste Kopfzeile"
14    varStory(11) = "Erste Fußzeile"
15    varStory(12) = "Fußnotentrennlinie"
16    varStory(13) = "Fußnoten-Fortsetzungstrennlinie"
17    varStory(14) = "Fußnoten-Fortsetzungshinweis"
18    varStory(15) = "Endnotentrennlinie"
19    varStory(16) = "Endnoten-Fortsetzungstrennlinie"
20    varStory(17) = "Endnoten-Fortsetzungshinweis"
21    If intStoryType < 1 Or intStoryType > UBound(varStory) Then
22        StoryTypeDescript = vbNullString
23    Else
24        StoryTypeDescript = varStory(intStoryType)
25    End If
26 End Function

```

Listing 3: VBA-Quellcode der Funktionsprozedur 'StoryTypeDescript'

² Quelle: Vgl. *Document.StoryRanges-Eigenschaft (Word)*, URL: <https://docs.microsoft.com/de-de/office/vba/api/word.document.storyranges>, gefunden am 18. Mai 2021

Komponente	Word-Objekt	Relation	Bemerkung	Beispiel
Haupttext	MainTextStory	1 pro Dokument	MainTextStory und Dokument können synonym behandelt werden.	ActiveDocument.Fields.Update
Hauptkopfzeile	PrimaryHeaderStory	1 pro Abschnitt (Section)	Ist immer vorhanden, kann aber auch leer sein. Kann auch der OddPageHeaderStory entsprechen, welche es nominell nicht gibt	ActiveDocument.Sections(1).Headers(wdHeaderFooterPrimary).Range.Fields.Update
Kopfzeile für erste Seite	FirstPageHeaderStory	0 oder 1 pro Abschnitt (Section)	Nur vorhanden, wenn im Seitenlayout "Erste Seite anders" aktiviert ist.	ActiveDocument.Sections(1).Headers(wdHeaderFooterFirstPage).Range.Fields.Update
Kopfzeile für gerade Seite	EvenPageHeaderStory	0 oder 1 pro Abschnitt (Section)	Nur vorhanden, wenn im Seitenlayout "Gerade-/ungerade Seiten anders" aktiviert ist.	ActiveDocument.Sections(1).Headers(wdHeaderFooterEvenPages).Range.Fields.Update
Hauptfusszeile	PrimaryFooterStory	1 pro Abschnitt (Section)	siehe Hauptkopfzeile	ActiveDocument.Sections(1).Footers(wdHeaderFooterPrimary).Range.Fields.Update
Fusszeile für erste Seite	FirstPageFooterStory	0 oder 1 pro Abschnitt (Section)	siehe oben	ActiveDocument.Sections(1).Footers(wdHeaderFooterFirstPage).Range.Fields.Update
Fusszeile für gerade Seite	EvenPageFooterStory	0 oder 1 pro Abschnitt (Section)	siehe oben	ActiveDocument.Sections(1).Footers(wdHeaderFooterEvenPages).Range.Fields.Update
Kommentare	CommentsStory	0 zu n pro Dokument	Kommentare können nur Objekte (Elemente) der MainTextStory sein	ActiveDocument.Comments(1).Range.Fields.Update
Endnote	EndnotesStory	0 oder 1 pro Dokument	Endnoten können nur Objekte (Elemente) der MainTextStory sein	ActiveDocument.Endnotes(1).Range.Fields.Update
Fussnoten	FootnotesStory	0 zu n pro Dokument	Fussnoten können nur Objekte (Elemente) der MainTextStory sein	ActiveDocument.Footnotes(1).Range.Fields.Update
Textfeld	TextFrameStory	0 zu n pro Story	Textfelder sind in erster Linie Objekte (Elemente) von Shapes	ActiveDocument.Shapes(1).TextFrame.TextRange.Fields.Update

Tabelle 1: Dokumentkomponenten (StoryTypes) in WORD³

³ Quelle: René Probst, Tabelle der «StoryTypes» in Word, URL: https://www.kastenmaier.de/wp-content/uploads/2007/07/Word_FAQ.pdf, gefunden am 24. Mai 2021