

Ausgewähltes Word-Dokument mit VBA-Prozeduren formal überprüfen

Dr. Volker Thormählen, 15. Mai 2020

Inhalt

1	Projekt 'CheckDocuments'	2
2	Prozeduren im Modul 'ReplaceFromTable'	2
3	Tabellen im Referenzdokument	3
3.1	Erste Tabelle	3
3.2	Zweite Tabelle	5
4	Bedeutung der Tabelleneinträge	7
5	Tabelleneinträge in dynamischen Arrays speichern	8
6	Bestimmte Dokumentkomponente durchsuchen	9
7	Prozeduren im Modul 'ScanDocParts'	10
8	Anwendungsbeispiel für das Suchen in Dokumentkomponenten	11
9	Anhang: VBA-Quellcode	12

Abbildungen

Abbildung 1:	Module im Projekt 'CheckDocuments'	2
Abbildung 2:	Struktogramm des Projekts 'ReplaceFromTable'	3
Abbildung 4:	Auszug aus Listing 10.	10
Abbildung 5:	Beispiel für Finden u. Ersetzen in Fußnoten ohne Benutzerdialog (links: vorher, rechts: nachher)	11
Abbildung 6:	Beispiel für Finden u. Ersetzen in Fußnoten mit Benutzerdialog (links: vorher, rechts: nachher)	11

Listings

Listing 1:	Prozedur 'ReplaceFromTable'	13
Listing 2:	Funktion 'fctShowFileDialogue'	14
Listing 3:	Funktion 'fctFindReplCount'	15
Listing 4:	Funktion 'fctHiLightTargets'	16
Listing 5:	Funktion 'fctFindWhiteSpace'	17
Listing 6:	Funktion 'fctExpandRange'	17
Listing 7:	Funktion 'fctGetCellContent'	17
Listing 8:	Prozedur 'RemoveHighlights'	17
Listing 9:	Funktion 'fctCheckFootnotes'	18
Listing 10:	Prozedur 'ScanDocParts'	19
Listing 11:	Funktion 'fctFRinStory'	20
Listing 12:	Funktion 'fctFindInStory'	21

Tabellen

Tabelle 1:	Name, Typ und Aufgabe der VBA-Prozeduren des Projekts	2
Tabelle 2:	Such- und Ersatzbegriffe	5
Tabelle 3:	Definitionen zur Prüfung der Interpunktion	6
Tabelle 4:	Deutsche bzw. englische Begriffe für das Suchen und Ersetzen	7
Tabelle 5:	Deutsche bzw. englische Begriffe für Interpunktionszeichen	8
Tabelle 6:	Word-Tastenkombinationen für Anführungszeichen	8
Tabelle 7:	Story Ranges (Dokumentkomponenten)	9

1 Projekt 'CheckDocuments'

Dieses Projekt betrifft das Suchen und Ersetzen in einem ausgewählten WORD-Dokument¹. Es beinhaltet zwei Module (s. Abbildung 1):

- *ReplaceFromTable*
- *ScanDocParts*

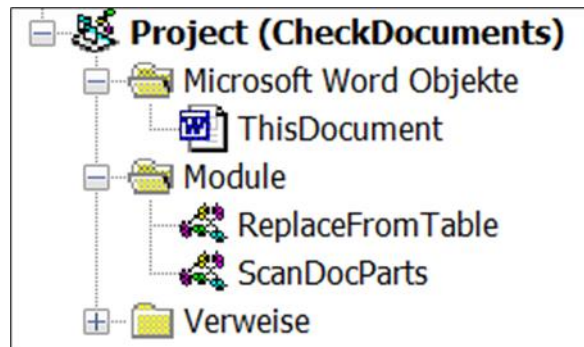


Abbildung 1: Module im Projekt 'CheckDocuments'

Die Prozeduren im Modul '*ReplaceFromTable*' behandeln die Aufgabe, wie ein ausgewähltes WORD-Dokument (sog. *Zieldokument*) auf formale Fehler untersucht und optional automatisch berichtigt werden kann. Die Anwendung basiert im Wesentlichen auf zwei Tabellen in einem weiteren Word-Dokument (sog. *Referenzdokument*).

- Die erste Tabelle im Referenzdokument dient zum Suchen und zum optionalen Ersetzen der Fundstellen.
- Die zweite Tabelle dient zur optionalen Hervorhebung von Fundstellen im Zieldokument. Möglicherweise erforderliche Ersetzungen müssen nachträglich manuell oder, falls verfügbar, mit anderen VBA-Prozeduren² vorgenommen werden.

Die Prozeduren im Modul '*ScanDocParts*' betreffen die Aufgabe, wie ausgewählte Dokumentkomponenten durchsucht und ggf. geändert werden können (s. Abschnitt 7).

2 Prozeduren im Modul 'ReplaceFromTable'

Das entsprechende Modul beinhaltet 2 Unterprozeduren und 7 Funktionsprozeduren (s. Tabelle 1):

Name der Prozedur	Typ	Aufgabe der Prozedur
ReplaceFromTable	sub	Finden und Ersetzen mit zwei Tabellen im Referenzdokument
RemoveHighlights	sub	Hervorhebungen im Zieldokument <i>optional</i> entfernen
fctShowFileDialoge	fct	Zieldokument mit Benutzerdialog auswählen
fctFindWhiteSpace	fct	Vorkommen des Zeichens (^w) auswerten
fctFindReplaceCount	fct	Benutzerdialog f. Suchen u. Ersetzen mit Rückgabe der Zahl der Ersetzungen
fctCheckFootnotes	fct	Fußnoten <i>optional</i> durchsuchen ggf. ersetzen
fctHiLightTargets	fct	Gefundene Suchbegriffe im Zieldokument <i>optional</i> gelb markieren
fctGetCellContent	fct	Inhalt (Text) einer Tabellenzelle bestimmen
fctExtendRange	fct	Fundstelle im Zieldokument auf ganzen Satz ausweiten

sub steht für Unterprozedur, **fct** steht für Funktionsprozedur

Tabelle 1: Name, Typ und Aufgabe der VBA-Prozeduren des Projekts

¹ Es wird empfohlen, die sog. *Dokumentprüfung* in WORD (Datei > Optionen > Dokumentprüfung) vorab durchzuführen.

² Zum Beispiel: Unpaarige Anführungszeichen und Klammern interaktiv korrigieren, <http://www.dr-thormaehlen.de/word/mismatchedpairs.pdf>

Abbildung 2 beinhaltet die Hierarchie der neun VBA-Prozeduren (vgl. Tabelle 1):

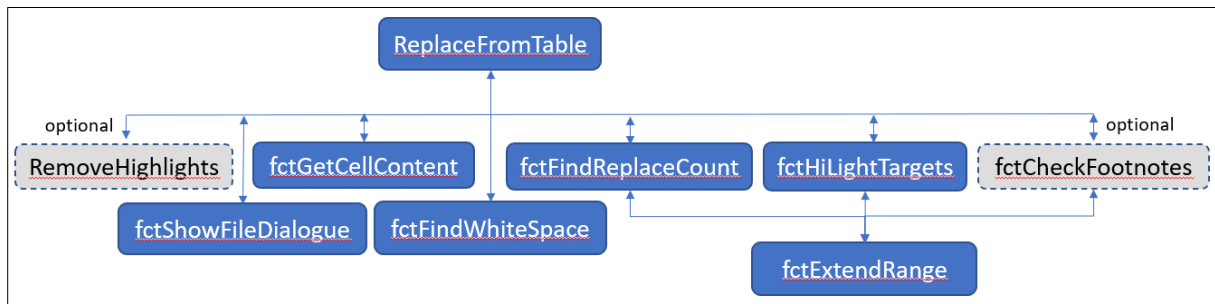


Abbildung 2: Struktogramm des Projekts 'ReplaceFromTable'

3 Tabellen im Referenzdokument

3.1 Erste Tabelle

Die erste Tabelle (s. Tabelle 2) besitzt 59 Zeilen mit jeweils drei Spalten. Die Spaltenköpfe sind mit *Suchen*, *Ersetzen* und *Beschreibung* überschrieben. Diese Tabelle wird von den Funktionsprozeduren mit dem Namen *fctFindReplaceCount* bzw. *fctCheckFootnotes* benötigt.

Suchen	Ersetzen	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;})([.,;:\?!\(\)\])	\2	1 oder mehr ungeschützte Leerzeichen vor Interpunktionszeichen bzw. öffnenden typografischen Anführungszeichen
([!])^0133	^32^0133	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)([.,;:\^\0145])	\1	Komma, Semikolon, Doppelpunkt o. Geviertstrich nach halbem öffnendem (d. h. rechtem) Anführungszeichen: Satzzeichen löschen!
(^\0146)([.,;:\^\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!
^=[]{1;}	^=	Halbgeviertstrich mit 1 oder mehr nachfolgenden Leerzeichen: Dito!
([])^0133	^32^0133	Fehlendes Leerzeichen vor horizontaler Ellipse: Einfügen!
^0133([])	^0133^32	Fehlendes Leerzeichen hinter horizontaler Ellipse: Einfügen!
^13{2;}	^p	2 oder mehr Absatzmarken in Folge: Eine erhalten!
^34{2;}	^34	2 oder mehr gerade doppelte Anführungszeichen in Folge: Eins erhalten!
^39{2;}	^39	2 oder mehr gerade halbe Anführungszeichen in Folge: Eins erhalten!
^l{1;}	^p	1 oder mehr manuelle Seitenumbrüche: Absatzmarke einfügen!
^13[]{1;}	^p	1 oder mehr Leerzeichen nach Absatzmarke: Diese erhalten! Siehe: ^p[^w]{1;}
^t{1;}	^t	1 oder mehr manuelle Zeilenumbrüche: Nur einen erhalten!
([0-9A-z])(^\0130)	\1^32\2	Alphanumerisches Zeichen direkt vor deutschem öffnendem typografischem halbem Anführungszeichen: Leerzeichen einfügen!
(.)(^\0130)	\1^32\2	Punkt, dito
(,)(^\0130)	\1^32\2	Komma, dito
([!])(^\0130)	\1^32\2	Ausrufezeichen, dito
([?])(^\0130)	\1^32\2	Fragezeichen, dito
([])(^\0130)	\1^32\2	Fehlendes Leerzeichen vor deutschen öffnenden typografischen halben Anführungszeichen: Leerzeichen einfügen!
(.)([.,;:\^\0145])	\2	Komma vor Punkt, Semikolon, Doppelpunkt, Halbgeviertstrich, Geviertstrich, Fragezeichen, Ausrufezeichen: Komma entfernen!
([.,;:\^\0145])(.)([])	\1	Komma nach Punkt, Semikolon, Doppelpunkt, Halbgeviertstrich, Geviertstrich, Fragezeichen, Ausrufezeichen: Komma entfernen!
([]{1;})([.,;:\?!\^\0145^\0147\)\}\]])	\2	Leerzeichen vor Satzzeichen, schließenden typograf. Anführungszeichen, runden, geschweiften bzw. eckigen Klammern entfernen

([.,;:\?!\^0145^0147\}\}\})([]{2;})	\1^32	Leerzeichen nach Satzzeichen, schließenden typograf. Anführungszeichen, runden, geschweiften bzw. eckigen Klammern entfernen
([A-z])(,)([A-z])	\1\2^32\3	Fehlendes Leerzeichen hinter Komma zwischen Satzteilen
([.\?!\])([A-Z])	\1^32\2	Fehlendes Leerzeichen hinter dem Ende (.?!) eines Satzes einfügen
([0-9])O([0-9])	\10\2	Großbuchstabe 'O' innerhalb einer Zahl durch Null (0) ersetzen
O([0-9])	0\1	Großbuchstabe 'O' vor einer Zahl durch Null (0) ersetzen
([0-9])O	\10	Großbuchstabe 'O' hinter einer Zahl durch Null (0) ersetzen
(^32)(-)(^32)	\2	Leerzeichen vor und nach Bindestrich entfernen
[^w]{1;}^p	^p	1 oder mehr Leerzeichen vor Absatzmarke: LZ entfernen
^p[^w]{1;}	^p	1 oder mehr Leerzeichen nach Absatzmarke: LZ entfernen

Tabelle 2: Such- und Ersatzbegriffe

3.2 Zweite Tabelle

Die zweite Tabelle (s. Tabelle 3) besitzt 52 Zeilen mit jeweils zwei Spalten. Die Spaltenköpfe sind überschrieben mit *Suchen* bzw. *Beschreibung*. Diese Tabelle wird von der VBA-Prozedur mit dem Namen *fctHiLightTargets* benötigt:

Suchbegriff	Beschreibung
.^s.^s.	Punkt vor geschütztem Leerzeichen ...
[\(]*\(\)	Beliebiger Text zwischen runden Klammern
[^02^34^39^=^+\\-]	Gerade (Apostroph, doppelte bzw. halbe Anführungszeichen, Halbgeviertstrich, Geviertstrich, Bindestrich)
[0-9A-z]^0130	Alphanummerischer Text vor dt. typografischem halben Anführungszeichen
.^0130	Punkt vor ...
,^0130	Koma vor ...
[\!]^0130	Ausrufezeichen vor ...
[\?]^0130	Fragezeichen vor ...
^0130;	Semikolon nach ...
^0130;	Doppelpunkt nach ...
^0130^+	Geviertstrich nach ...
^+^0130	Geviertstrich vor ...
^0145^0130	Schließende vor öffnender ...
[0-9A-z]^0132	Alphanummerischer Text vor
.^0132	Punkt vor...
,^0132	Komma vor...
[\!]^0132	Ausrufezeichen vor ...
[\?]^0130	Fragezeichen vor ...
[0-9A-z]^0132	Beliebiger Text vor dt. öffnenden typografischen doppelten Anführungszeichen
.^0132	Punkt vor ...
,^0132	Komma vor ...
[\!]^0132	Ausrufezeichen vor ...
[\?]^0132	Fragezeichen vor ...
^0132;	Semikolon nach ...
^0132:	Doppelpunkt nach ...
^0132^+	Geviertstrich nach ...
^+^0132	Geviertstrich vor ...
^0147^0132	Schließendes vor öffnendes ...
[^13]^0145[0-9A-z]	Dt. typografisches schließendes halbes Anführungszeichen nach Absatzmarke
^+^0145^+	Geviertstrich nach dt. typografischen schließenden halben Anführungszeichen
^+^0145	Geviertstrich vor dt. typografischen schließenden halben Anführungszeichen
^0130^0145	Öffnende vor schließenden dt. typografischen halben Anführungszeichen
^0147[0-9A-z]	Dt. typografische öffnende doppelte Anführungszeichen vor beliebigem Text
^0147^+	Geviertstrich nach dt. typografischen öffnenden doppelten Anführungszeichen
^+^0147	Geviertstrich vor ...
^s^0145[!^s]	Geschütztes Leerzeichen vor dt. schließendem halben Anführungszeichen
^=	Halbgeviertstrich
^+	Geviertstrich
^39{1;}	1 oder mehr gerade halbe Anführungsstriche in Folge
„	Komma vor Punkt
„,	Komma nach Punkt
„;	Semikolon vor Komma
„;	Semikolon nach Komma
„:	Komma vor Doppelpunkt
„;	Komma nach Doppelpunkt
„[\?],	Komma vor Fragezeichen
[\?],	Komma nach Fragezeichen
„[\!]	Koma vor Ausrufezeichen
[\!],	Komma nach Ausrufezeichen
„[:;\?\\!]	Komma vor Satzzeichen
[:;\?\\!],	Komma nach Satzzeichen

Tabelle 3: Definitionen zur Prüfung der Interpunktion

4 Bedeutung der Tabelleneinträge

Zum Verständnis der Einträge in den oben erwähnten Tabellen (s. Tabelle 2 u. Tabelle 3) kann der Inhalt der Tabelle 4 beitragen:

Deutsch Name	Englischer Name	Suchen	Ersetzen
Irgendein Zeichen	Any character	^?	
Irgendeine Zahl	Any digit	^#	
Irgendein Buchstabe	Any letter	^\$	
Caret Zeichen	Caret character	^^	^^
Inhalt der Zwischenablage	Clipboard contents		^c
Spaltenumbruch	Column break	^n	^n
'Finde welchen Text' (was auch immer während der Suche gefunden wurde)	'Find what text' (whatever was found during the search)		^&
Geviertstrich	Em dash (Alt+0151)	^+	^+
Halbgeviertstrich	En dash (Alt+0150)	^=	^=
Endnotenzeichen	Endnote mark	^e	
Feld	Field	^d	
Fußnotenzeichen	Footnote mark	^f	
Grafik	Graphic	^g	
Zeilenumbruch	Line break	^l	^l
Manueller Seitenumbruch	Manual page break	^m	^m
Geschützter Bindestrich	Nonbreaking hyphen	^~	^~
Geschütztes Leerzeichen	Nonbreaking space	^s	^s
Optionalen Bindestrich	Optional hyphen	^-	^-
Absatzmarke (-zeichen)	Paragraph mark	^p	^p
Abschnittsumbruch	Section break	^b	
Tabstoppezeichen	Tab character	^t	^t
Leerzeichen	White Space	^w	

Tabelle 4: Deutsche bzw. englische Begriffe für das Suchen und Ersetzen

Einige Zeichen wie [\ ^ \$. | ? ! * + () sind *„speziell“* und müssen *„maskiert“* werden, bevor sie wirksam verwendet werden können. Um einem solchen Zeichen zu *„entkommen“*, muss davor ein umgekehrter Schrägstrich (\) (engl. backslash) eingesetzt werden³. Um beispielsweise das Suchen für 2 oder mehr Fragezeichen in Folge erfolgreich auszuführen, muss das Feld *„Suchen nach“* wie folgt belegt werden: (\?){2;}. Außerdem ist die Suchoption *„Platzhalter verwenden“* zu aktivieren⁴.

Tabelle 5 enthält deutsche bzw. englische Begriffe für *Interpunktionszeichen*.

Beim Finden und Ersetzen von *Anführungszeichen* sind die Einträge in Tabelle 6 hilfreich.

³ Eingabe mit der Tastatur: Alt + 92 oder mit der in WORD eingebauten Textfunktion: Chr(92).

⁴ In Deutschland ist das im System eingestellte Listentrennzeichen i. d. R. ein Semikolon, also *nicht* das Komma.

Zeichen	englischer Name	deutscher Name
.	period	Punkt
?	question mark	Fragezeichen
!	exclamation point	Ausrufezeichen
,	comma	Komma
;	semicolon	Semikolon
:	colon	Doppelpunkt
-	em dash	Geviertstrich
-	en dash	Halbgeviertstrich
-	hyphen	Bindestrich/Minus
[]	brackets	eckige Klammern
{}	braces	geschwungene Klammern
()	parantheses	runde Klammern
'	apostrophe	Apostroph (gerade)
"..."	quotation marks	doppelte Anführungszeichen
'...'	single quotation marks	halbe Anführungszeichen
...	ellipsis	Auslassungszeichen

Tabelle 5: Deutsche bzw. englische Begriffe für Interpunktionszeichen

Deutsch/engl. Anführungszeichen		Zeichen	links (öffnend)	rechts (schließend)
gerade doppelte ¹	straight double	„...“	Alt+0034	Alt+0034
gerade halbe (einfache) ¹	straight single	'...'	Alt+0039	Alt+0039
typografische doppelte (normale), dt.	curly (smart) double	„...“	Alt+0132	Alt+0147
typografische doppelte (normale), engl.		„...”	Alt+0147	Alt+0148
typografische halbe (einfache), dt.	curly (smart) single	'...'	Alt+0130	Alt+0145
typografische halbe (einfache), engl.		'...'	Alt+0145	Alt+0146
doppelte, spitze	double angle (frz. guillemets)	»...«	Alt+0187	Alt+0171
einfache, spitze	single angle	›...‹	Alt+0155	Alt+0139
¹ Gerade doppelte bzw. gerade halbe Anführungszeichen brauchen <i>nicht</i> maskiert zu werden, weil sie zum ASCII-Zeichensatz gehören.				

Tabelle 6: Word-Tastenkombinationen für Anführungszeichen

5 Tabelleneinträge in dynamischen Arrays speichern

Um die Laufzeit der Prozeduren zu verkürzen, werden die Einträge in Tabelle 2 bzw. Tabelle 3 jeweils in sog. *dynamischen Arrays* gespeichert (s. Listing 1).

- Dim arrFind() As String ' dyn. Array für Suchbegriffe
- Dim arrReplace() As String ' dyn. Array für Ersatzbegriffe

Dabei wird die Funktionsprozedur *fctGetCellContent* (s. Abbildung 2) ausgerufen.

Die genannten Arrays werden von den betreffenden Funktionsprozeduren herangezogen, also nicht mehr die gerade genannten Tabellen im Referenzdokument.

6 Bestimmte Dokumentkomponente durchsuchen

Um zu erreichen, dass nicht nur im sog. *Hauptdokumentbereich* gesucht und ggf. gefunden wird, sondern zusätzlich in best. Dokumentkomponenten, muss das Modul '*ReplaceFromTable*' (s. Abbildung 2) entsprechend erweitert werden⁵. Diese Erweiterungsmöglichkeit wird hier zunächst am Beispiel der Dokumentkomponente *Fußnoten* (s. Tabelle 7, 3. Zeile) ausführlich gezeigt (s. Listing 9).

Story Range	Nr.	Dokumentkomponente
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	Fußnoten-Fortsetzungstrennlinie
wdFootnoteContinuationNoticeStory	14	Fußnoten-Fortsetzungshinweis
wdEndnoteSeparatorStory	15	Endnotentrennlinie
wdEndnoteContinuationSeparatorStory	16	Endnoten-Fortsetzungstrennlinie
wdEndnoteContinuationNoticeStory	17	Endnoten-Fortsetzungshinweis

Tabelle 7: Story Ranges (Dokumentkomponenten)

Quelle: <https://docs.microsoft.com/de-de/dotnet/api/microsoft.office.interop.word.wdstorytype?view=word-pia>

⁵ Webseiten im Internet, die den Suchbegriff '*For Each rngStory In ActiveDocument.StoryRanges*' enthalten, helfen häufig weiter.

7 Prozeduren im Modul 'ScanDocParts'

Zum Projekt 'CheckDocuments' (s. Abbildung 1) gehört das Modul 'ScanDocParts'. Dazu zählen folgende Prozeduren (s. Tabelle 8):

Name der Prozedur	Typ	Aufgabe der Prozedur
ScanDocParts	sub	Best. Dokumentkomponente(n) im Zieldokument durchsuchen.
fctFRinStory	fct	Ersetzung(en) in best. Dokumentkomponente(n) vornehmen und zählen (<i>ohne</i> Benutzerdialog)
fctFindInStory	fct	Ersetzung(en) in best. Dokumentkomponente(n) vornehmen und zählen (<i>mit</i> Benutzerdialog)

sub steht für Unterprozedur **fct** steht für Funktionsprozedur

Tabelle 8: Name, Typ und Aufgabe der VBA-Prozeduren im Modul 'ScanDocParts'

Prozedur 'ScanDocParts' (s. Listing 10) und die dazugehörigen Funktionsprozeduren 'fctFRinStory' (s. Listing 11) bzw. 'fctFindInStory' (s. Listing 12) sind zum Suchen und Ersetzen in ausgewählten Dokumentkomponenten eines bestimmten Zieldokuments ohne/mit Benutzerdialog geeignet. Dafür ist eine Schleife notwendig,

- die alle Dokumentkomponenten des Zieldokuments (docTgt) durchläuft,
- dabei die gewünschten Dokumentkomponenten auswählt
- und dort die erforderlichen Änderungen bzw. Markierungen vornimmt (vgl. Listing 10).
Abbildung 3 enthält einen Auszug daraus.

```

For Each rngStory In docTgt.StoryRanges
|   ' Hauptdokument bzw. Fußnoten auswählen
|   Select Case rngStory.StoryType
|   |   ' Hier die Dokumentkomponente(n) auswählen
|   |   Case wdMainTextStory, wdFootnotesStory
|   |   |   ' Hier die Funktionsprozedur zum Suchen u. Ersetzen einfügen.
|   |   Case Else
|   End Select
Next rngStory

```

Abbildung 3: Auszug aus Listing 10

8 Anwendungsbeispiel für das Suchen in Dokumentkomponenten

Angenommen, im Zieldokument (docTgt) befindet sich eine Fußnote mit dem Text "Heute ist ein schöner Tag.". Diese Fußnote soll geändert werden in: "Heute ist ein wunderschöner Tag." Such- bzw. Ersatzbegriff enthalten somit folgende Zeichenketten:

```
strFind = "schöner"           ' Suchbegriff
strReplace = "wunderschöner" ' Ersatzbegriff
```

Das Ausführen der Prozedur 'ScanDocParts' (s. Listing 10) mit der Funktionsprozedur 'fctFRinStory' (s. Listing 11) führt zu folgendem Ergebnis (s. Abbildung 4):

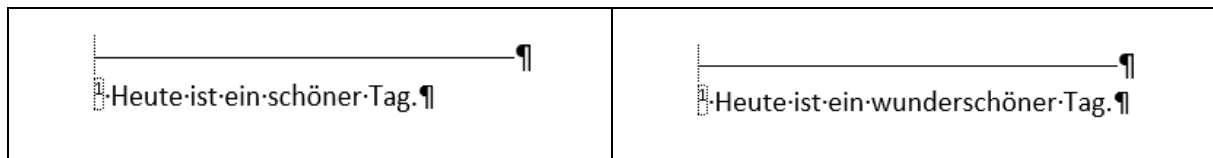


Abbildung 4: Beispiel für Finden u. Ersetzen in Fußnoten **ohne** Benutzerdialog (links: vorher, rechts: nachher)

Nach Einbau eines Benutzerdialogs in die Funktionsprozedur ergibt sich (s. Abbildung 5):

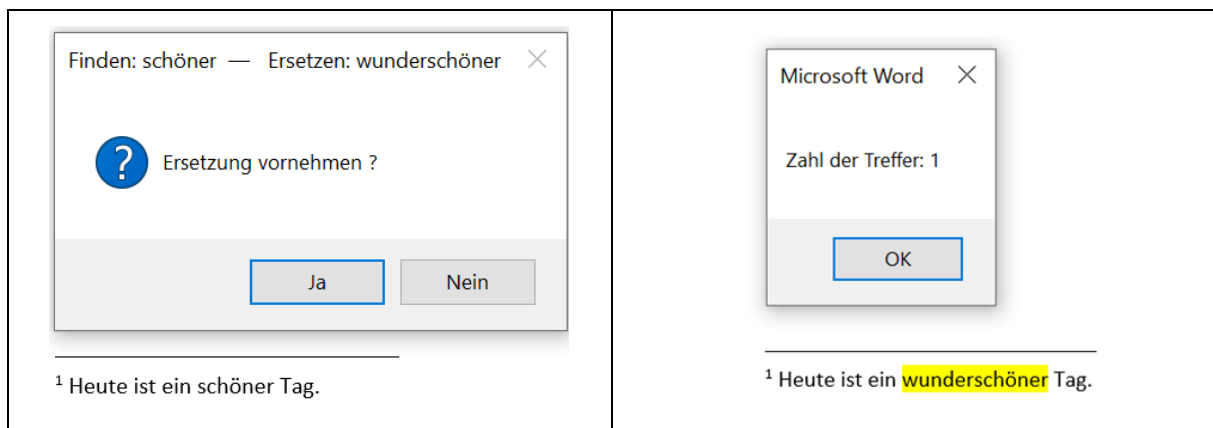


Abbildung 5: Beispiel für Finden u. Ersetzen in Fußnoten **mit** Benutzerdialog (links: vorher, rechts: nachher)

9 Anhang: VBA-Quellcode

```

Option Explicit
Option Base 1

Sub ReplaceFromTable()
' Aufgabe: Finden u. Ersetzen mit dyn. Arrays
Const conTitle As String = "ReplaceFromTable"
Dim blnSwitch As Boolean ' boolescher Schalter f. MatchWildcards
Dim docRef As Document ' Referenzdokument
Dim docTgt As Document ' Zieldokument
Dim lngLoop As Long ' Schleifenzähler
Dim lngHits As Long ' Trefferzähler
Dim lngRow As Long ' Zeile der Tabelle
Dim lngRowsCnt As Long ' Zeilenzahl der Tabelle
Dim lngIdx As Long ' Index für dyn. Datenfelder
Dim lngColor As Long ' Hervorhebungsfarbe
Dim objTbl As Table ' Tabelle
Dim objCell As Cell ' Tabellenzelle
Dim rngFind As Range ' Suchbegriff
Dim rngRepl As Range ' Ersatzbegriff
Dim strFileNm As String ' Dateiname des Zieldokuments
Dim arrFind() As String ' dyn. Array für Suchbegriffe
Dim arrReplace() As String ' dyn. Array für Ersatzbegriffe

On Error GoTo ErrorHandler ' Fehlerbehandlung

' Bildschirmaktualisierung einschalten wg. Änderungen im Zieldokument
Application.ScreenUpdating = True

' Referenzdokument mit zwei Tabellen bestimmen
Set docRef = ActiveDocument
With docRef
' 1. Tabelle auswählen
Set objTbl = docRef.Tables(2)
End With

' Gesamte Zeilenzahl der Tabelle bestimmen
lngRowsCnt = objTbl.Rows.Count

' Dyn. Arrays dimensionieren
ReDim arrFind(lngRowsCnt - 1)
ReDim arrReplace(lngRowsCnt - 1)

' Dyn. Arrays mit Such- bzw. Ersatzbegriffen belegen
lngIdx = 0
For lngRow = 2 To lngRowsCnt
lngIdx = lngIdx + 1
Set objCell = objTbl.Cell(lngRow, 1)
arrFind(lngIdx) = fctGetCellContent(objCell)
Set objCell = objTbl.Cell(lngRow, 2)
arrReplace(lngIdx) = fctGetCellContent(objCell)
Next lngRow

' Zieldokument mit Benutzerdialog auswählen
strFileNm = fctShowFileDialogue
If Len(strFileNm) > 0 Then ' Zieldokument öffnen
Set docTgt = Documents.Open(FileName:=strFileNm, ReadOnly:=False, _
AddToRecentFiles:=False, Visible:=True)
Else
MsgBox Prompt:="Abbruch: Auswahl der Zieldatei ist fehlgeschlagen!", _
Buttons:=vbExclamation, Title:=conTitle
GoTo ExitPoint
End If

' Hervorhebungen im Zieldokument optional entfernen
Call RemoveHighlights(docTgt, conTitle)

' Farbe zum Hervorheben speichern und festlegen
lngColor = Options.DefaultHighlightColorIndex
Options.DefaultHighlightColorIndex = wdYellow

For lngLoop = 1 To UBound(arrFind)
blnSwitch = fctFindWhiteSpace(arrFind(lngLoop))

```

```

docTgt.Activate
lngHits = lngHits + fctFindReplCount(docTgt, _
    arrFind(lngLoop), arrReplace(lngLoop), blnSwitch)
' Optionale Ergänzung des Projekts f. Fußnoten
If MsgBox(Prompt:="Fußnoten prüfen?", _
    Buttons:=vbYesNo + vbQuestion, _
    Title:=conTitle) = vbYes Then
    lngHits = lngHits + fctCheckFootnotes(docTgt, _
        arrFind(lngLoop), arrReplace(lngLoop), blnSwitch)
End If
Next lngLoop

' Erfolgsmeldung ausgeben
MsgBox Prompt:=CStr(lngHits) & " Ersetzung(en) vorgenommen!", _
    Buttons:=vbExclamation, Title:=docTgt.Name

' Zweite Phase starten mit der 3. Tabelle im Referenzdokument
Set objTbl = docRef.Tables(3)

' Gesamte Zeilenzahl der 3. Tabelle bestimmen
lngRowsCnt = objTbl.Rows.Count

' Dyn. Array für Suchbegriffe dimensionieren
ReDim arrFind(lngRowsCnt - 1)

' Dyn. Array mit Tabelleneinträgen belegen
lngIdx = 0
For lngRow = 2 To lngRowsCnt
    lngIdx = lngIdx + 1
    Set objCell = objTbl.Cell(lngRow, 1)
    arrFind(lngIdx) = fctGetCellContent(objCell)
Next lngRow

' Relevante Zeilen der 2. Tabelle durchlaufen
For lngLoop = 1 To UBound(arrFind)
    ' Gefundene Suchbegriffe optional gelb markieren.
    lngHits = lngHits + fctHiLightTargets(docTgt, arrFind(lngLoop))
Next lngLoop

' Erfolgsmeldung ausgeben
MsgBox CStr(lngHits) & " Fundstellen gelb markiert!", vbExclamation

ExitPoint:
On Error Resume Next
' Farbe zum Hervorheben zurücksetzen
Options.DefaultHighlightColorIndex = lngColor

' Referenzdokument unverändert schließen
docRef.Close SaveChanges:=wdDoNotSaveChanges, OriginalFormat:=wdOriginalDocumentFormat
Set docRef = Nothing
Set objTbl = Nothing

' Zieldokument speichern und anzeigen
With docTgt
    If .ReadOnly = False Then
        If Not .Saved Then .Save
    End If
    .Activate
End With
ErrorHandler:
MsgBox "Laufzeitfehler " & Err.Number & ", " & Err.Description, vbExclamation, conTitle
Resume ExitPoint
End Sub

```

Listing 1: Prozedur 'ReplaceFromTable'

```
Function fctShowFileDialog()  
    ' Aufgabe: Zieldokument mit Benutzerdialog auswählen.  
    Dim fd As FileDialog ' Dateidialogfeld  
    Set fd = Application.FileDialog(msoFileDialogFilePicker)  
    With fd  
        .AllowMultiSelect = False  
        .Title = "Word-Datei auswählen"  
        .InitialFileName = Options.DefaultFilePath(wdDocumentsPath) & "\*.docx"  
        .ButtonName = "Auswählen"  
        With .Filters  
            .Clear  
            .Add "Word-Dateien", "*.docx; *.docm", 1  
            .Add "Alle Dateien", "*.*"  
        End With  
        .Show  
        If .SelectedItems.Count = 0 Then  
            fctShowFileDialog = vbNullString  
        Else  
            fctShowFileDialog = .SelectedItems(1)  
        End If  
    End With  
    Set fd = Nothing  
End Function
```

Listing 2: Funktion 'fctShowFileDialog'

```

Private Function fctFindReplCount( _
    ByRef docTgt As Document, _
    ByVal strFind As String, _
    ByVal strRepl As String, _
    ByVal blnSwitch As Boolean) As Long
    ' Aufgabe: Benutzerdialog f. Suchen u. Ersetzen mit Rückgabe d. Zahl d. Ersetzungen.
    ' Argumente: docTgt ' Zieldokument
    '           strFind ' Suchbegriff
    '           strRepl ' Ersetzung
    '           blnSwitch ' boolescher Schalter f. MatchWildcards
    Dim objRng As Range ' Textbereich
    Dim lngCount As Long ' Zähler f. Ersetzungen
    Dim strText As String ' Text in Zeile der Fundstelle (Umfeld der Fundstelle)
    With docTgt
        .Activate
        ' Textbereich am Anfang des Zieldokuments bestimmen
        Set objRng = .Range(0, 0)
    End With
    ' Anfangswert für Zähler der Ersetzungen setzen
    fctFindReplCount = 0
    With objRng.Find
        .ClearFormatting
        .Text = strFind
        With .Replacement
            .ClearFormatting
            .Text = strRepl
            .Highlight = True
        End With
        .Forward = True
        .Wrap = wdFindStop
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchAllWordForms = False
        .MatchSoundsLike = False
        .MatchWildcards = blnSwitch
        .Execute
        Do While .Found = True
            ' Fundstelle ausweiten f. Meldung im Dialogfeld
            strText = fctExpandRange(objRng)
            strText = strText & Space(1) & Chr(133) & vbCrLf & vbCrLf
            Select Case MsgBox(Prompt:=strText & "Ersetzung durchführen?", _
                Buttons:=vbYesNoCancel, _
                Title:="Finden: " & strFind & " " & Chr(151) & " Ersetzen: " & strRepl)
                Case vbYes
                    .Execute Replace:=wdReplaceOne
                    objRng.HighlightColorIndex = wdYellow
                    ' Zähler f. Ersetzungen erhöhen
                    lngCount = lngCount + 1
                Case vbCancel
                    Exit Do ' Scheife beenden
                Case Else: ' nichts tun
            End Select
            objRng.Collapse Direction:=wdCollapseEnd
            .Execute
        Loop
    End With
    ' Zähler d. Ersetzungen zurückgeben
    fctFindReplCount = lngCount
End Function

```

Listing 3: Funktion 'fctFindReplCount'

```

Private Function fctHiLightTargets(ByRef docTgt As Document, ByVal strFind As String) As Long
' Aufgabe: Gefundene Suchbegriffe im Zieldokument optional gelb markieren.
' Argumente: docTgt ' Zieldokument
'           strFind ' Suchbegriff
Dim objRng As Range ' Textbereich
Dim lngCount As Long ' Zahl der Markierungen
Dim lngHits As Long ' Zahl der Treffer
Dim strText As String ' Satz der Fundstelle

fctHiLightTargets = 0 ' Anfangswert setzen
With docTgt
    .Activate
    Set objRng = .Range(0, 0)
End With
With objRng.Find
    .ClearFormatting
    .Text = strFind
    With .Replacement
        .Text = "^&"
        .Highlight = True
    End With
    .Forward = True
    .Wrap = wdFindStop
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchAllWordForms = False
    .MatchSoundsLike = False
    .MatchWildcards = True
    .Execute
    Do While .Found = True
        ' Trefferzahl erhöhen
        lngHits = lngHits + 1
        ' Fundstelle auswählen
        objRng.Select
        ' Fundstelle ausweiten f. Meldung im Dialogfeld
        strText = fctExpandRange(objRng)
        strText = strText & Space(1) & Chr(133) & vbCrLf & vbCrLf
        Select Case MsgBox(Prompt:=strText & " Fundstelle gelb markieren?", _
            Buttons:=vbYesNoCancel + vbQuestion, _
            Title:="Finden: " & strFind)
            Case vbYes
                objRng.HighlightColorIndex = wdYellow
                lngCount = lngCount + 1
            Case vbCancel
                Exit Do ' Schleife beenden
            Case Else ' nichts tun
        End Select
        objRng.Collapse Direction:=wdCollapseEnd
        .Execute
    Loop
End With
' Zahl der Hervorhebungen zurückgeben
fctHiLightTargets = lngCount
End Function

```

Listing 4: Funktion 'fctHiLightTargets'


```

Private Function fctFindWhiteSpace(ByVal strFind As String) As Boolean
    ' Aufgabe: Vorkommen des Zeichens (^w) auswerten.
    ' ^w steht für Leerraum/Leerzeichen/Tabulator/etc. (⇒ auch Weißraum genannt)
    If InStr(strFind, "^w") > 0 Then
        fctFindWhiteSpace = False
    Else
        fctFindWhiteSpace = True
    End If
End Function

```

Listing 5: Funktion 'fctFindWhiteSpace'

```

Private Function fctExpandRange(objRng As Range) As String
    ' Aufgabe: Fundstelle auf ganzen Satz ausweiten.
    fctExpandRange = vbNullString
    ' Fundstelle auswählen
    objRng.Select
    ' Fundstelle markieren
    objRng.HighlightColorIndex = wdTurquoise
    ' Fundstelle aufbereiten f. Meldung im Dialogfeld
    With Selection
        .StartOf Unit:=wdSentence, Extend:=wdMove
        .MoveEnd Unit:=wdSentence, Count:=1
        ' Expandierten Textbereich zurückgeben
        fctExpandRange = .Text
    End With
End Function

```

Listing 6: Funktion 'fctExpandRange'

```

Function fctGetCellContent(ByRef objCell As word.Cell) As String
    ' Aufgabe: Zelleninhalt bestimmen
    Dim objRng As Range
    Set objRng = objCell.Range
    objRng.End = objRng.End - 1
    fctGetCellContent = objRng.Text
    Set objRng = Nothing
End Function

```

Listing 7: Funktion 'fctGetCellContent'

```

Private Sub RemoveHighlights(docTgt As Document)
    ' Aufgabe: Alle Hervorhebungen im Zieldokument entfernen.
    If MsgBox("Hervorhebungen im Zieldokument entfernen", vbYesNo + vbQuestion) = vbYes Then
        docTgt.Range.HighlightColorIndex = wdNoHighlight
        With docTgt
            If .Saved = False Then .Save
        End With
    End If
End Sub

```

Listing 8: Prozedur 'RemoveHighlights'

```

Public Function fctCheckFootnotes(docTgt As Document, strFind As String, strRepl As String, blnSwitch As
Boolean) As Long
' Aufgabe: Gefundene Suchbegriffe in den Fußnoten optional gelb markieren.
' Argumente: docTgt ' Zieldokument
' strFind ' Suchbegriff
' strRepl ' Ersatzbegriff
' blnSwitch ' boolescher Schalter f. MatchWildcards
Dim objRng As Range ' Suchbereich
Dim lngCount As Long ' Zahl der Markierungen
Dim lngHits As Long ' Zahl der Treffer
Dim strText As String ' Satz der Fundstelle
fctCheckFootnotes = 0 ' Anfangswert der Funktion setzen
If docTgt.Footnotes.Count > 0 Then
    Set objRng = docTgt.StoryRanges(wdFootnotesStory) ' Dokumentkomponente: => Fußnote
    objRng.Collapse Direction:=wdCollapseStart
    With objRng.Find
        .ClearFormatting
        .Text = strFind
        With .Replacement
            .ClearFormatting
            .Text = strRepl
            .Highlight = True
        End With
        .Forward = True
        .Wrap = wdFindStop
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchAllWordForms = False
        .MatchSoundsLike = False
        .MatchWildcards = blnSwitch
        .Execute
    Do While .Found = True
        ' Fundstelle ausweiten f. Meldung im Dialogfeld
        strText = fctExpandRange(objRng)
        strText = strText & Space(1) & Chr(133) & vbCrLf & vbCrLf
        Select Case MsgBox(Prompt:=strText & "Ersetzung durchführen?", Buttons:=vbYesNoCancel, _
            Title:="Finden: " & strFind & " " & Chr(151) & " Ersetzen: " & strRepl)
            Case vbYes
                .Execute Replace:=wdReplaceOne
                objRng.HighlightColorIndex = wdYellow
                ' Zähler f. Ersetzungen erhöhen
                lngCount = lngCount + 1
            Case vbCancel
                Exit Do ' Scheife beenden
            Case Else: ' nichts tun
        End Select
        objRng.Collapse Direction:=wdCollapseEnd
        .Execute
    Loop
    End With
Else
    MsgBox "Keine Fußnoten im Dokument " & docTgt.Name & " gefunden", vbExclamation
End If
fctCheckFootnotes = lngCount
End Function

```

Listing 9: Funktion 'fctCheckFootnotes'

```

Public Sub ScanDocParts()
    Dim docTgt As Document ' Zieldokument
    Dim lngCount As Long ' Trefferzähler
    Dim lngRetVal As Long ' Rückgabewert d. MsgBox
    Dim rngStory As Range ' Dokumentkomponente
    Dim strFileName As String ' Dateiname d. Zieldokuments
    Dim strFind As String ' Suchbegriff
    Dim strReplace As String ' Ersatzbegriff

    On Error GoTo ErrorHandler ' Fehlerbehandlung

    ' Zieldokument öffnen
    strFileName = "C:\Users\volker\Documents\MeinTestDoc.docx"
    Set docTgt = Documents.Open(FileName:=strFileName, _
        ReadOnly:=False, AddToRecentFiles:=False)

    strFind = "schöner" ' Suchbegriff
    strReplace = "wunderschöner" ' Ersatzbegriff
    lngCount = 0 ' Anfangswert f. Trefferzähler

    ' Abfrage: Ersetzen mit Benutzerdialog?
    retVal = MsgBox(Prompt:="Ersetzen mit Benutzerdialog?", _
        Buttons:=vbYesNoCancel + vbQuestion, _
        Title:="In Dokumentkomponente(n) ersetzen.")
    If retVal = vbCancel Then
        MsgBox "Abbruch d. Benutzer!", vbOKOnly + vbExclamation
        Goto ExitPoint
    End If

    With docTgt
        .Activate
        ' Alle Markierungen im Zieldokument entfernen
        .Range.HighlightColorIndex = wdNoHighlight
    End With

    ' Alle Dokumentkomponente(n) im Zieldokument durchlaufen
    For Each rngStory In docTgt.StoryRanges
        Select Case rngStory.StoryType
            ' Haupttext und Fußnoten auswählen
            Case wdMainTextStory, wdFootnotesStory
                Select Case retVal
                    Case vbYes ' mit Benutzerdialog
                        lngCount = lngCount + fctFndInStory(rngStory, _
                            strFind, strReplace, False)
                    Case vbNo ' ohne Benutzerdialog
                        lngCount = lngCount + fctFRinStory(rngStory, _
                            strFind, strReplace, False)
                End Select
            Case Else
                End Select
        Next rngStory
    With docTgt
        If .Saved = False Then .Save
        .Activate
    End With
    MsgBox "Normales Ende. Zahl der Treffer: " & CStr(lngCount)
ExitPoint:
    Exit Sub
ErrorHandler:
    If Err.Number = 5174 Then
        MsgBox "Abbruch: Zieldokument nicht gefunden!", _
            vbOKOnly + vbExclamation, "Dateiname: " & strFileName
    Else
        MsgBox "Laufzeitfehler: " & CStr(Err.Number) & ", " & _
            Err.Description, vbOKOnly, "ScanDocParts"
    End If
    Resume ExitPoint
End Sub

```

Listing 10: Prozedur 'ScanDocParts'

```
Private Function fctFRinStory(rngStory As Range, _  
    strFind As String, _  
    strReplace As String, _  
    blnSwitch As Boolean) As Long  
    ' Aufgabe: Ersetzung(en) in best. Dokumentkomponente vornehmen und zählen (ohne Benutzerdialog).  
    Dim lngCount As Long          ' Trefferzähler  
    rngStory.Collapse Direction:=wdCollapseStart  
    With rngStory.Find  
        .ClearFormatting  
        .Replacement.ClearFormatting  
        .Text = strFind  
        .MatchWildcards = blnSwitch  
    Do While .Execute  
        With rngStory  
            .Text = strReplace          ' Ersetzung vornehmen  
            .Collapse Direction:=wdCollapseEnd  
        End With  
        lngCount = lngCount + 1  
    Loop  
    End With  
    fctFRinStory = lngCount  
End Function
```

Listing 11: Funktion 'fctFRinStory'

```
Private Function fctFindInStory(rngStory As Range, _
    strFind As String, _
    strReplace As String, _
    bInSwitch As Boolean) As Long
' Aufgabe: Ersetzung(en) in best. Dokumentkomponente(n) vornehmen und zählen (mit Benutzerdialog)
Dim lngCount As Long ' Anfangswert f. Trefferzähler
With rngStory.Find
Do While .Execute(FindText:=strFind, Forward:=True, MatchWildcards:=bInSwitch) = True
    rngStory.Select
    If MsgBox(Prompt:="Ersetzen?", _
        Buttons:=vbYesNo + vbQuestion, _
        Title:="Finden: " & strFind & Space(2) & Chr(151) & Space(2) & _
        " Ersetzen: " & strReplace) = vbYes Then
        With rngStory
            .Text = strReplace ' Ersetzung vornehmen
            .HighlightColorIndex = wdYellow ' Fundstelle gelb markieren
            .Collapse Direction:=wdCollapseEnd
        End With
        lngCount = lngCount + 1
    End If
Loop
End With
fctFindInStory = lngCount ' Rückgabewert der Funktion
End Function
```

Listing 12: Funktion 'fctFindInStory'