

Praktische Beispiele für erweitertes Suchen und Ersetzen mit Platzhaltern in Word-Dokumenten

Dr. Volker Thormählen, 14. Mai 2017

Inhaltsverzeichnis	Seite
1 Einleitung	3
2 Suchen und Ersetzen mit Mustervergleich	4
2.1 Alle Leerzeichen vor einer Absatzendemarke entfernen	4
2.2 Mit Bindestrich getrennte Wörter wieder verbinden	4
2.3 Vor- und Nachnamen vertauschen	4
2.4 Alle Zeichen zwischen Anführungszeichen kursiv formatieren	5
2.5 Öffnende und schließende Anführungszeichen auswechseln	5
3 Tabelle mit Platzhaltern für die Erweiterte Suche	6
4 Benutzerformular	8
4.1 Aufruf	8
4.2 Layout	8
4.3 Steuerelemente	9
4.4 Befehlsschaltflächen	10
5 Ablauf der Prozeduren	10
5.1 Eingaben	10
5.2 Ausgaben	11
5.3 Ersetzungen	11
6 Klick-Prozeduren	11
6.1 Zieldatei auswählen	11
6.2 Prozedur starten	13
6.3 Prozedur abbrechen	13
6.4 Formular schließen	14
7 Abbruch der laufenden Hauptprozedur	14
8 Hauptprozedur	15
8.1 Suche mit Mustervergleich im Zieldokument durchführen	17
8.2 Ablauf der Hauptprozedur verzögern	19
9 E-Mail-Link	19
10 Laufzeiten	19
11 Grenzen	20
12 Anhang	20

13	Abbildungsverzeichnis.....	22
14	Tabellenverzeichnis.....	22
15	Quellcodelistings.....	22
16	Literaturverzeichnis.....	22

1 Einleitung

In diesem Beitrag werden zahlreiche praktische Beispiele für das Suchen und Ersetzen mit Platzhaltern in einem Word-Dokument aufgeführt. Mit Platzhaltern können in einem Word-Dokument Zeichenfolgen gesucht werden, die ein bestimmtes *Suchmuster* repräsentieren. Damit können formale Mängel in einem Word-Dokument gefunden werden, die sonst vielleicht unentdeckt bleiben würden. Der Beitrag wendet sich also in erster Linie an Verfasser und Korrekturleser umfangreicher Word-Dokumente, die diese mit Hilfe der sog. „Erweiterten Suche“ formal berichtigen wollen.

Wer lange Dokumente verfasst und/oder diese Korrektur liest kann das Ergebnis seiner Arbeit qualitativ verbessern, wenn die in MS Word eingebauten Funktionalitäten bestmöglich genutzt werden. Neben „Rechtschreibung und Grammatik“ und „Formatvorlagen“ zählt dazu insbesondere die Funktion zum Suchen und Ersetzen (Navigation: Start ► Bearbeiten ► Suchen bzw. Ersetzen). Bei der sog. „Erweiterten Suche“ kann zum Beispiel die Suchoptionen „Platzhalter verwenden“ markiert werden. Unterhalb des Eingabefelds „Suchen nach“ erscheint daraufhin der Hinweis „Mit Mustervergleich“. Dementsprechend muss das Eingabefeld u. a. mit ausgewählten Platzhalterzeichen syntaktisch richtig belegt werden, um im Zieldokument Zeichenfolgen zu suchen und möglicherweise zu finden, die das vorgegebene Suchmuster¹ repräsentieren.

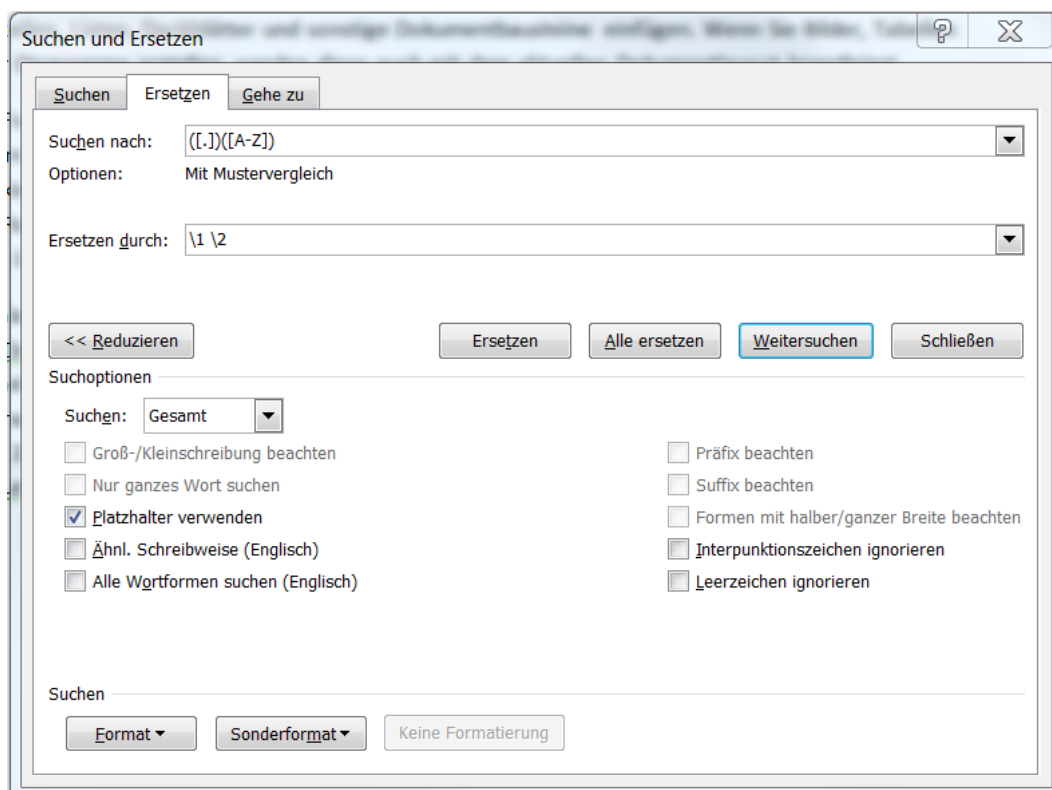


Abbildung 1: Suchen und Ersetzen mit Platzhaltern

Im Folgenden werden zunächst 5 repräsentative Beispiele für das Suchen und Ersetzen mit Platzhaltern in Word-Dokumenten präsentiert.

Danach wird eine Word-Tabelle gezeigt, ...

¹ Leerzeichen einfügen, wenn es hinter einem Satzzeichen (hier: Punkt) fehlt.

1. die gültige Einträge für das Eingabefeld „Suchen nach“ enthält,
2. jeweils dazugehörige Eingaben für das Feld „Ersetzen durch“ ausweist und
3. die Aufgabe für vorstehende Tabelleneinträge kurz beschreibt.

Im Weiteren wird ein Word-Benutzerformular gezeigt, das dazu dient, die oben genannten Tabelleneinträge auf ein ausgewähltes Word-Dokument (sog. *Zieldokument*) anzuwenden.

Schließlich werden relevante VBA²-Prozeduren, die bei der Verarbeitung der Tabelleneinträge zum Einsatz kommen, aufgelistet und erklärt.

2 Suchen und Ersetzen mit Mustervergleich

2.1 Alle Leerzeichen vor einer Absatzendemarke entfernen

- *Suchen nach:* ([]{1;})(^13) Mit Mustervergleich
- *Ersetzen durch:* \2 oder: ^p

Erläuterung:

Unter „Erweitern“ muss „Platzhalter verwenden“ aktiviert sein. Der Wert in geschweiften Klammern {1;} gibt an, wie oft ein Leerzeichen vorkommen darf, hier mindestens 1-mal. ^13 und ^p repräsentieren jeweils eine Absatzendemarke.

In das Feld „Ersetzen durch“ wird \2 (Treffer des 2. runden Klammerpaars) oder ^p eingefügt, mit anderen Worten, die führenden Leerzeichen werden entfernt, die Absatzendemarke selbst bleibt erhalten.

2.2 Mit Bindestrich getrennte Wörter wieder verbinden

Um mit Bindestrich getrennte Wörter (z. B. "Muster- suche“) zusammenzufügen, sind folgende Eingaben erforderlich:

- *Suchen nach:* ([a-z])- []{1}
- *Ersetzen durch:* \1

Erläuterung:

Es wird nach einem beliebigen Kleinbuchstaben [a-z] gesucht, gefolgt von 1 Bindestrich und 1 Leerzeichen. Der Wert in geschweiften Klammern {1} gibt an, wie oft der Bindestrich und das Leerzeichen vorkommen dürfen, hier genau 1-mal. Nur der gefundene Kleinbuchstabe bleibt erhalten, die beiden folgenden Zeichen werden entfernt. Unter „Erweitern“ muss wiederum „Platzhalter verwenden“ markiert sein.

2.3 Vor- und Nachnamen vertauschen

Eine Liste mit Vor- und Nachnamen soll umgestellt werden: Nachname zuerst, gefolgt von Komma und Leerzeichen, dann der jeweilige Vorname.

Listeneinträge vor der Umstellung:

² VBA ist die Abkürzung für **V**isual **B**asic for **A**pplications, eine Skriptsprache vom *Microsoft* für die *Office*-Suite.

Jens Schmidt
Günter Müller
Armin Schneider

- *Suchen nach:* (<*>) (<*>) Mit Mustervergleich
- *Ersetzen durch:* \2, \1

Erläuterung:

Die beiden runden Klammerpaare im Feld „Suchen nach“ beinhalten jeweils:

- < Zeichen für Wortanfang
- * Platzhalter für beliebige Zeichen einschl. Interpunktion
- > Zeichen für Wortende

Wichtig: Zwischen den beiden Ausdrücken (<*>) muss sich ein Leerzeichen befinden!

Im Feld „Ersetzen durch“ wird zwischen den beiden Ausdrücken ein Komma eingefügt, gefolgt von einem Leerzeichen.

Listeneinträge nach der Umstellung:

Schmidt, Jens
Müller, Günter
Schneider, Armin

2.4 Alle Zeichen zwischen Anführungszeichen kursiv formatieren

Alle Zeichen zwischen doppelten geraden (sog. Gänsefüßchen) oder doppelten typografischen Anführungszeichen sollen *kursiv* formatiert werden.

Text vor der Ersetzung: Er sagte: „Alles ganz toll.“ Sie erwiderte: "Das finde ich auch."

- *Suchen nach:* [^0034|^0132]*[^0034|^0147] Mit Mustervergleich
- *Ersetzen durch:* ^& Schriftart: Kursiv

Text nach der Ersetzung: Er sagte: „*Alles ganz toll.*“ Sie erwiderte: "*Das finde ich auch.*"

Erläuterung:

^0nnn repräsentiert ein ANSI³ Zeichen, wobei nnn die numerische Nummer⁴ des Zeichens darstellt.

Der senkrechte Strich (|) repräsentiert den logischen Operator „oder“.

Das Stern-Zeichen (*) repräsentiert beliebige Zeichen.

^& repräsentiert im Feld „Ersetzen durch“ den Inhalt des Feldes „Suchen nach“.

2.5 Öffnende und schließende Anführungszeichen auswechseln

Alle Zeichen zwischen spitzen Anführungszeichen (sog. Guillemets oder Chevrons) sollen stattdessen mit doppelten typografischen Anführungszeichen umschlossen werden.

Text vor der Ersetzung: »Hier steht ein Zitat«.

- *Suchen nach:* (»)(*)(«) oder: (^0187)(*)(^0171) Mit Mustervergleich
- *Ersetzen durch:* „\2“ oder: ^0132\2^0147

³ ANSI ist die Abkürzung für *American National Standard Institute*.

⁴ Mit der Tastenkombination **Alt** + [Ziffer mit vorangestellter 0] können die ANSI-Zeichen aufgerufen werden.

Text nach der Ersetzung: „Hier steht ein Zitat“.

Erläuterung:

^0nnn repräsentiert ein ANSI Zeichen, wobei nnn die numerische Nummer des Zeichens darstellt.

Das Suchen und Ersetzen mit Mustervergleich in einem Word-Dokument setzt voraus, dass der Benutzer

- die zulässigen Platzhalterzeichen kennt
- und weiß oder probiert, wie diese zu einem gültigen Ausdruck für das jeweilige Eingabefeld kombiniert werden können.

Weitere praktische Beispiele werden im Folgenden tabellarisch aufgelistet, jedoch nicht im Einzelnen erläutert. Zum Thema „Suchen und Ersetzen mit Platzhaltern in Word-Dokumenten“ bieten diese Fachartikel einen detaillierten Überblick: [1], [2].

3 Tabelle mit Platzhaltern für die Erweiterte Suche

Die Einträge in Tabelle 1 werden für die „Erweiterte Suche“ mit Platzhaltern verwendet. Sie dient als externe Datenquelle für die weiter unten beschriebene die Hauptprozedur „ReadAndUpdateTable“ (s. Listing 8: Hauptprozedur

Die Ausdrücke können aber auch ohne das Word-Benutzerformular direkt in die Maske zur „Erweiterte Suche“ (s. Abbildung 1) eingetragen werden.

Die Einträge in der Tabellenspalte „Suchen nach“ repräsentieren Suchmuster, die syntaktisch richtig definiert sind. Vgl. dazu [1] und [2]. Die meisten davon eignen sich für die Suche nach formalen Fehlern in einem ausgewählten Word-Zieldokument, die dort sonst nur schwer und/oder zeitaufwändig zu finden sind.

Suchen nach	Ersetzen durch	Beschreibung der Aufgabe
([!^13^0147.;\?!\!])(^13)	\1<period>\2	<i>insert missing period (dot) before paragraph break</i> Fehlendes Satzendezeichen (Punkt) vor Absatzendezeichen einfügen.
([.\?!\!])([A-z])	\1<space>\2	<i>insert missing space after sentence</i> Fehlendes Leerzeichen nach einem Satzende einfügen.
([.,;\?!\!])([]{2;})	\1<space>	<i>remove excess spaces behind punctuation mark</i> <i>Überflüssige Leerzeichen nach Satzzeichen entfernen.</i>
([.,;\?!\!])([! ^s^13])	\1<space>\2	<i>Insert missing space behind punctuation mark</i> <i>Fehlendes Leerzeichen nach Satzzeichen einfügen.</i>
(^2)([.,;\?!\!])	\2\1	<i>move punctuation mark in front of footnote number</i> Satzzeichen vor die Fußnotennummer verschieben.
([.,;\?!\!])(^2)	\2\1	<i>move punctuation mark behind footnote number</i> Satzzeichen hinter die Fußnotennummer verschieben.
([]{1;})(^2)	\2	<i>remove all spaces before footnote number</i> Alle Leerzeichen vor einer Fußnotennummer entfernen.
([<^0132])(*)([>^0147>])	^&	<i>reformat text inside curly quotes</i> Text zwischen typografischen Anführungszeichen suchen und ggf. neu formatieren (fett, kursiv, ...), d. h. nur das Format und nicht der Text der Fundstelle soll ersetzt werden.
(\d)(*)(\d)	\1\2\3 oder ^&	<i>reformat text inside square brackets</i> Text zwischen eckigen Klammern neu formatieren.
([.,;])(-)	\1	<i>remove hyphen after punctuation</i> Gedankenstrich hinter Satzzeichen entfernen.

<code>([.\?\\!])[^s]</code>	<code>\1<space></code>	<i>remove non-breaking (that is, protected) space after punctuation</i> Geschütztes Leerzeichen nach Satzendezeichen durch ein normales ersetzen.
<code>([A-Za-z])-(){1,}([A-Za-z])</code>	<code>\1\2\4</code>	<i>remove multiple spaces after hyphen</i> Multiple Leerzeichen nach einem Bindestrich entfernen.
<code>(){1,}([;:])</code>	<code>\2</code>	<i>remove multiple spaces before punctuation marks</i> Multiple Leerzeichen vor Satzzeichen entfernen.
<code>([0-9]{1,})-([0-9]{1,})</code>	<code>\1^=\3</code>	<i>replace hyphens between number parts with en-dash (nut)</i> Bindestrich zwischen Nummernteilen durch Halbgeviertstrich ersetzen.
<code>([0-9])-([0-9])</code>	<code>\1^=\2</code>	<i>replace hyphens between numbers with 1 en-dash</i> Bindestrich zwischen Nummern durch 1 Halbgeviertstrich ersetzen
<code>([.\?\\!])^(0147)</code>	<code>\2\1</code>	<i>move closing curly quote in front of punctuation mark</i> Schließendes typografisches Anführungszeichen vor Satzendezeichen verschieben.
<code>^(0147)([.\?\\!])</code>	<code>\2\1</code>	<i>move punctuation mark in front of closing curly quote</i> Satzendezeichen vor schließendes typografisches Anführungszeichen verschieben.
<code>(\)\)\)(){1,}([.\?\\!])</code>	<code>\1\3</code>	<i>remove multiple spaces after closing brackets</i> Multiple Leerzeichen nach einer schließenden Klammer entfernen.
<code>([A-Za-z])(){2,}([A-Za-z])</code>	<code>\1<space>\3</code>	<i>remove extra space between words</i> Überflüssige Leerzeichen zwischen Wörtern entfernen.
<code>([^s^t]{1,})^(13)</code> <code>([^t^l^s]{1,})^(13{1,})</code>	<code>\2</code>	<i>remove multiple „white spaces“ before paragraph breaks</i> Alle Leerstellen vor einer Absatzendemarke entfernen.
<code>^(13)([^s^t]{1,})"</code> <code>^(13{1,})([^t^l^s]{1,})</code>	<code>\1</code>	<i>remove multiple „white spaces“ behind paragraph breaks</i> Alle Leerstellen nach einer Absatzendemarke entfernen.
<code>(){2,}</code>	<code><space></code>	<i>remove double spaces</i> Überflüssige Leerzeichen entfernen.
<code>[]([]{1,})@([!])</code>	<code><space>\2</code>	<i>remove excess spaces</i> Überflüssige Leerzeichen entfernen.
<code>^13{2,}</code>	<code>^p</code>	<i>stray empty paragraphs</i> Leere Absätze entfernen.
<code>^13 []{1,}</code>	<code>^p</code>	<i>remove hanging spaces before a paragraph</i> Leerzeichen am Anfang eines Absatzes entfernen.
<code>[]{1,}^13</code>	<code>^p</code>	<i>remove trailing spaces at the end of a paragraph</i> Leerzeichen am Ende eines Absatzes entfernen
<code>([!])([()])</code>	<code>\1<space>\2</code>	<i>insert missing space before opening bracket</i> Fehlendes Leerzeichen vor öffnender runder Klammer einfügen.
<code>^(0132)(*)(^0147)([.\?\\!])</code>	<code>\1\2\4\3</code>	<i>swap end punctuation mark before closing curly quote</i> Satzendezeichen vor schließendes Anführungszeichen verschieben.
<code>([.\?\\!])([A-Z])</code>	<code>\1<space>\2</code>	<i>insert missing space after sentence</i> Fehlendes Leerzeichen nach Satzende einfügen.
<code>([!])(\([()])</code>	<code>\1<space>\2</code>	<i>insert missing space before opening parenthesis</i> Fehlendes Leerzeichen vor öffnender runder Klammer einfügen.
<code>(\)\)\)([! ,;.\?\\!])</code>	<code>\1<space>\2</code>	<i>insert missing space after closing parenthesis</i> Fehlendes Leerzeichen nach schließender runder Klammer einfügen.
<code>(){2,}([1-9])</code>	<code><space>\2</code>	<i>remove multiple spaces before a number</i> Multiple Leerzeichen vor einer Nummer entfernen.
<code>[.]{3}</code>	<code>^0133</code>	<i>replace 3 periods (dots) with 1 ellipsis character</i> 3 Auslassungspunkte durch 1 Auslassungszeichen ersetzen.
<code>[!]^0133</code>	<code><space>^0133</code>	<i>insert missing space before the ellipsis character</i> Fehlendes Leerzeichen vor Auslassungszeichen einfügen.
<code>^0133[!]</code>	<code>^0133<space></code>	<i>insert missing space after the ellipsis character</i> Fehlendes Leerzeichen nach Auslassungszeichen einfügen.
<code>[-]{3}</code>	<code>^+</code>	<i>replace 3 consecutive hyphens with 1 em-dash (mutton)</i> 3 Bindestriche in Folge durch 1 Geviertstrich ersetzen.
<code>[-]{2}</code>	<code>^=</code>	<i>replace 2 consecutive hyphens with 1 en-dash (nut)</i> 2 Bindestriche in Folge durch 1 Halbgeviertstrich ersetzen.
<code>([a-z])-(){1,}([A-Z])</code>	<code>\1\3</code>	<i>remove space behind hyphen between nouns</i> Leerzeichen entfernen bei Bindestrich zwischen Hauptwörtern.

<[A-Z]{2;}>	^&	<i>search for consecutive capital letters (that is, find acronyms)</i> Suche nach Großbuchstabe in Folge (d. h. Akronyme finden).
(,)([! 0-9^13])	\1<space>\2	<i>search for a comma without a space after it</i> Komma suchen ohne Leerzeichen danach.
(:)([! ^13//])	\1<space>\2	<i>search for a colon without a space after it</i> Doppelpunkt suchen ohne Leereichen danach.
^13([a-z]{1})	^&	<i>correct paragraphs that start with no capitalization</i> Absätze berichtigen, die mit Kleinschreibung beginnen. (Format: Großbuchstaben)
(<[A-Za-z]{1;}> \1[!A-Za-z]	\1<space>	<i>find and remove duplicate words</i> Doppelte Wörter finden und entfernen.
([^s]{1;})([.,;:\?!\])	\2	<i>remove all spaces in front of punctuation marks</i> Alle Leerzeichen vor Satzzeichen entfernen.

Hinweis: In der zweiten Spalte dieser Tabelle repräsentiert <space> ein Leerzeichen im „Ersetzen durch“ Feld. Entsprechend repräsentiert <period> einen Punkt.
Die Suche nach multiplen Leerzeichen ([]{1;}) bzw. ([]{2;}) kommt in der Tabelle jeweils viermal vor, aber in einem anderen Zusammenhang.

Tabelle 1: Einträge für die sog. Erweiterte Suche in einem Word-Dokument

4 Benutzerformular

Das Benutzerformular wird benutzt, um ...

- ein Zieldokument interaktiv zu öffnen,
- alle Einträge in Tabelle 1 für die Suche mit Platzhaltern im Zieldokument zu benutzen,
- Tabelleneinträge fett zu markieren, die bei der Suche zu Treffern geführt hat.

4.1 Aufruf

Das Benutzerformular (Name: *frmControl*) wird automatisch aufgerufen, sobald das Quelldokument geöffnet wird:

```
Private Sub Document_Open()
    ' Benutzerformular 'frmControl' ungebunden aufrufen
    frmControl.Show vbModeless
End Sub
```

Listing 1: Benutzerformular ungebunden aufrufen

4.2 Layout

Der Aufbau des Benutzerformulars ergibt sich aus Abbildung 2. Im Bereich „Eingabe“ sind alle Steuerelemente deaktiviert, außer dem Textfeld „Wartezeit ...“ und dem Kontrollkästchen „*englisch*“.

Das Textfeld „Ersetzen durch“ wird nur der Vollständigkeit halber angezeigt, es dient zurzeit nicht zum automatischen Ersetzen.

Alle Steuerelemente in den Bereichen „Ausgabe“ und „Kontrolle“ sind deaktiviert. Ihre Belegung erfolgt jeweils automatisch mittels VBA-Prozedur.

Zum Schließen auf die Befehlsschaltfläche rechts unten klicken!

MS WORD: Praktische Beispiele für 'Erweiterte Suche' mit Platzhaltern

Eingabe

Vollständiger Name der Zieldatei:

Anzahl Seiten: Wartezeit nach jedem Suchmuster : Sekunden

Zeile: Zeilen: Suchen nach (Suchmuster):

englisch:

Platzhalter:

Ausgabe

Trefferzahl: Fehlermeldung:

Kontrolle

Zeichen vorher: Zeichen nachher:

Laufzeit: hh:mm:ss

E-Mail: volker@dr-thormaehlen.de

Abbildung 2: Layout des Benutzerformulars (Name: frmControl)

4.3 Steuerelemente

Das Benutzerformular (s. Abbildung 2) besteht aus insgesamt 40 Steuerelementen⁵. Tabelle 2 enthält die Namen und Beschriftungen aller Steuerelemente. Ihre Namen dienen zur Auswertung des Benutzerformulars durch VBA-Routinen.

Befehlsschaltflächen	Beschriftung
cmdCancel	Prozedur abbrechen
cmdClose	Formular schließen
cmdGetTgtFile	Zieldatei auswählen
cmdStart	Prozedur starten
Bezeichnungsfelder	Inhalt
lblContol	Kontrolle
lblEMail	E-Mail:
lblHyperlink	www.dr-thormaehlen.de
lblInput	Eingabe
lblOutput	Ausgabe
lblRunTime	hh:mm:ss
lblRunTm	Laufzeit

⁵ frmControl.controls.count = 40

lblTitle	MS WORD: Praktische Beispiele für erweitertes Suchen ...	
	Beschriftung	Kontrollkästchen
	englisch	chkEnglisch
	Platzhalter	chkPlaceholder
Bezeichnungsfelder	Inhalt	Textfelder
lblCharCount1	Zeichen vorher	txtCharCount1
lblCharCount2	Zeichen nachher	txtCharCount2
lblElapsedTm	Laufzeit	txtElapsedTm
lblError	Fehlermeldung	txtError
lblHits	Zahl der Treffer	txtHits
lblPages	Anzahl Seiten	txtPages
lblPattern	Suchen nach (Suchmuster)	txtPattern
lblReplace	Ersetzen durch	txtReplace
lblRow	Zeile	txtRow
lblRows	Zeilen	txtRows
lblTask	Aufgabe	txtTask
lblTgtNm	Vollständiger Name der Zieldatei	txtTgtNm
lblWaitTm	Wartezeit nach jedem Suchmuster	txtWaitTm
<i>Bedeutung der Präfixe:</i>		
lbl = label, txt = text, chk = check box		

Tabelle 2: Steuerelemente des Benutzerformulars

4.4 Befehlsschaltflächen

Das Benutzerformular ist mit 4 Befehlsschaltflächen bestückt, s. Tabelle 2, oben:

- *Zieldatei auswählen:* Beim Anklicken dieser Schaltfläche öffnet sich ein Word eingebautes Dialogfeld zur Auswahl einer sog. Zieldatei. Bei erfolgreicher Auswahl werden der Name der Zieldatei sowie die Seitenzahl des entsprechenden Dokuments im Benutzerformular automatisch angezeigt.
- *Hauptprozedur starten:* Diese Schaltfläche ist anfangs nicht aktiviert. Die Aktivierung geschieht erst nach erfolgreicher Auswahl der Zieldatei.
- *Hauptprozedur abbrechen:* Damit kann der Benutzer die zuvor gestartete Prozedur beenden.
- *Formular schließen:* Das Benutzerformular kann sofort nach seinem Aufruf wieder geschlossen werden, sonst nur bei Abbruch oder Ende der gestarteten Prozedur.

5 Ablauf der Prozeduren

5.1 Eingaben

Die Prozedur erhält die erforderlichen Eingabedaten aus einer externen Word-Tabelle, s. Tabelle 1. Davon gibt es drei Ausnahmen:

- Die Wartezeit⁶ nach jedem Suchmuster kann geändert werden.
- Die Bezeichnung der Aufgabe eines Suchmusters kann von deutsch (Standardwert) nach englisch geändert werden.
- Der Inhalt des Kontrollkästchens „Platzhalter“ ist fest vorgegeben.

⁶ Vorbelegt mit 2 Sekunden.

Angezeigt, aber derzeit nicht verwendet, wird der Inhalt der 2. Tabellenspalte mit der Überschrift „Ersetzen durch“. Die Begründung dafür ist weiter unten zu finden (s. 5.3).

5.2 Ausgaben

Die Prozedur durchläuft die Tabelleneinträge (s. Tabelle 1) mit einer sog. *Do-While*-Schleife und ermittelt durch den Aufruf einer gesonderten VBA-Funktion die Zahl der Fundstellen. Ist die Trefferzahl positiv, wird

- der zugehörige Eintrag in der 1. Spalte der externen Word-Tabelle *fett* formatiert und
- die *Trefferzahl* im Benutzerformular angezeigt.

Ggf. wird auch eine Fehlermeldung im Benutzerformular ausgegeben.

5.3 Ersetzungen

Zurzeit nimmt die VBA-Prozedur bei einem Treffer keine automatische Ersetzung vor, weil

- die Genauigkeit des Treffers fehlen könnte und
- der Kontext der Fundstelle nicht überschaubar sein könnte.

Programmtechnisch ist der Einbau der automatischen Ersetzung ziemlich einfach zu bewerkstelligen, indem die Fundstellen in einer *Do-While*-Schleife durchlaufen und jeweils abgefragt wird, ob und wie ersetzt werden soll.

PRAXISTIPP: Beim erweiterten Suchen und Ersetzen mit Platzhaltern ist generell davon abzuraten, die Option „Alle ersetzen“ bedenkenlos zu wählen.

6 Klick-Prozeduren

Folgende vier Klick-Prozeduren sind mit den entsprechenden Befehlsschaltflächen (s. Abbildung 2 und Tabelle 2) verbunden:

Klick-Prozedur	Aufgabe
cmdGetTgtFile_Click	Zieldatei auswählen und Anfangswerte im Benutzerformular setzen
cmdStart_Click	Hauptprozedur starten
cmdCancel_Click	Hauptprozedur abbrechen
cmdClose_Click	Benutzerformular schließen

Der jeweils dazugehörige VBA-Quellcode wird in den folgenden vier Listings wiedergegeben. Sie enthalten jeweils Kommentare und Hervorhebungen, um das Verständnis zu erleichtern.

6.1 Zieldatei auswählen

Die Zieldatei wird interaktiv ausgewählt, s. Listing 2:

```
Private Sub cmdGetTgtFile_Click()
Private Sub cmdGetTgtFile_Click()
    ' Aufgabe: Zieldatei auswählen und Anfangswerte im Benutzerformular setzen.
    Dim docTgt      As Document ' Zieldokument
    Dim lngChar1    As Long     ' Zeichenzahl im Zieldokument
    Dim strFileNm   As String   ' Dateiname des Zieldokuments
    On Error GoTo Err_Point    ' Fehlerbehandlung
    strFileNm = fctShowFileDialog ' Zieldatei im Dialog auswählen
    If Len(strFileNm) > 0 Then
```

```

' Zieldokument öffnen
Set docTgt = Documents.Open(FileName:=strFileNm, _
    ReadOnly:=True, AddToRecentFiles:=False, Visible:=False)
' Anfängliche Zahl der Zeichen im Zieldokument ermitteln
lngChar1 = docTgt.Range.Characters.Count
With Me
    .txtTgtNm = strFileNm          ' Dateiname des Zieldokuments
    ' Seitenzahl im Zieldokument
    .txtPages = docTgt.Range.Information(wdNumberOfPagesInDocument)
    .txtCharCount1 = lngChar1      ' anfängliche Zeichenzahl im Zieldokument ermitteln
    .txtElapsedTm.Text = "00:00:00" ' Anfangswert für Laufzeit setzen
    .txtWaitTm.SetFocus
    .cmdStart.Enabled = True
    .cmdCancel.Enabled = False
    .cmdGetTgtFile.Enabled = False
End With
Else
    MsgBox "Fehler bei der Auswahl der Zieldatei", vbCritical, "cmdGetTgtFile_Click"
End If
Exit_Point:
Exit Sub
Err_Point:
MsgBox "Laufzeitfehler: " & Err.Description, vbCritical, "cmdGetTgtFile_Click"
Resume Exit_Point
End Sub

```

Listing 2: Klick-Prozedur zur Auswahl des Zieldokuments und zur anfänglichen Belegung der Felder des Benutzerformulars

Die in Listing 2 gelb hervorgehobene Code-Zeile `strFileNm = fctShowFileDialogue` dient dazu, den vollständigen Name der im Dialog ausgewählten Zieldatei in der Variablen `strFileNm` zu speichern.

Mit der Funktion `fctShowFileDialogue` wird die Dateiauswahl gesteuert, s. Listing 3:

```

Function fctShowFileDialogue()
' Aufgabe: Word-Datei in Dialog mit dem Benutzer auswählen
Dim fd As FileDialog ' Dateialogfeld
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
        fctShowFileDialogue = vbNullString
    Else
        fctShowFileDialogue = .SelectedItems(1)
    End If
End With
Set fd = Nothing
End Function

```

Listing 3: Funktion zur Auswahl einer Word-Datei im Dialog

6.2 Prozedur starten

Die Klick-Prozedur `cmdStart_Click` (s. Listing 4)

- setzt den Anfangswert (False) des Abbruchschalters für die Hauptprozedur,
- speichert die Startzeit der Hauptprozedur,
- ruft dann die Hauptprozedur auf und
- berechnet danach deren Laufzeit.

```
Private Sub cmdStart_Click()  
    ' Aufgabe: Hauptprozedur aufrufen  
    Dim intHours      As Integer  ' Stunden  
    Dim intMinutes    As Integer  ' Minuten  
    Dim intSeconds    As Integer  ' Sekunden  
    Dim lngElapsedTm  As Long     ' Laufzeit  
    Dim strStartTm    As String   ' Startzeit  
    Dim strEndTm      As String   ' Endezeit  
    bolCancel = False           ' Abbruchschalter  
    With frmControl  
        .cmdCancel.Enabled = True  
        .cmdStart.Enabled = False  
        .cmdClose.Enabled = True  
    End With  
    strStartTm = Format(Now, "hh:mm:ss") ' Startzeit speichern  
    frmControl.cmdClose.Enabled = False  
    ' Externe Tabelle durchlaufen und dabei das Benutzerformular "frmControl" aktualisieren  
    Call ReadAndUpdateTable ' Hauptprozedur aufrufen  
    strEndTm = Format(Now, "hh:mm:ss") ' Endezeit speichern  
    lngElapsedTm = DateDiff("s", strStartTm, strEndTm) ' Laufzeit ermitteln  
    intHours = Int(lngElapsedTm / 3600)  
    intMinutes = Int((lngElapsedTm Mod 3600) / 60)  
    intSeconds = lngElapsedTm Mod 60  
    With frmControl  
        ' Laufzeit im Format hh:mm:ss ausgeben  
        .txtElapsedTm.Text = Format(intHours, "00") & ":" & _  
                               Format(intMinutes, "00") & ":" & _  
                               Format(intSeconds, "00")  
        .cmdClose.Enabled = True  
    End With  
End Sub
```

Listing 4: Klick-Prozedur starten und Hauptprozedur aufrufen

Die in Listing 4 gelb hervorgehobene Code-Zeile `Call ReadAndUpdateTable` ruft die Hauptprozedur mit ihrem Namen auf. Den VBA-Code der Hauptprozedur dokumentiert Listing 8: Hauptprozedur

6.3 Prozedur abbrechen

Die logische Variable `cmdCancel` erhält beim Abbrechen den Wert „wahr“, s. Listing 5.

```
Private Sub cmdCancel_Click()  
    ' Aufgabe: Laufende Hauptprozedur abbrechen  
    bolCancel = True  
End Sub
```

6.4 Formular schließen

Diese Klick-Prozedur schließt alle offenen Word-Dokumente, aber nicht das Zieldokument.

```
Private Sub cmdClose_Click()
    ' Aufgabe: Benutzerformular schließen
    Dim intLoop As Integer ' Schleifenzähler
    Dim strKeepDocOpen As String ' Dateiname des Zieldokuments
    ' Alle offenen Dokumente schließen, aber nicht das aktuelle Dokument
    strKeepDocOpen = ActiveDocument.Name
    For intLoop = Documents.Count To 1 Step -1
        If Documents(intLoop).Name <> strKeepDocOpen Then
            Documents(intLoop).Close SaveChanges:=wdDoNotSaveChanges
        End If
    Next intLoop
    ' Aufgabe: Benutzerformular und Word-Dateien schließen
    Dim intLoop As Integer ' Schleifenzähler
    Dim strKeepDocOpen As String ' Dateiname des Zieldokuments
    ' Alle offenen Dokumente schließen, aber nicht das aktuelle Zieldokument
    strKeepDocOpen = ActiveDocument.Name
    For intLoop = Documents.Count To 1 Step -1
        If Documents(intLoop).Name <> strKeepDocOpen Then
            Documents(intLoop).Close SaveChanges:=wdDoNotSaveChanges
        End If
    Next intLoop
    Unload Me ' Benutzerformular schließen
End Sub
```

Listing 6: Klick-Prozedur zum Schließen des Benutzerformulars

7 Abbruch der laufenden Hauptprozedur

Die Programmierlogik zum Abbrechen der laufenden Hauptprozedur wird in Listing 7 gezeigt:

```
Option Explicit
' Logische Variablen auf Prozedurebene deklarieren
Public bolCancel As Boolean ' Abbrechen
Public bolRetVal As Boolean ' Rückgabewert

Private Sub cmdStart_Click()
    bolCancel = False ' Anfangswert setzen
    ...
End Sub

Private Sub cmdCancel_Click()
    ' Aufgabe: Hauptprozedur abbrechen
    bolCancel = True
End Sub

Public Sub ReadAndUpdateTable() ' Hauptprozedur
    ...
    Do While lngRow <= lngLastRow
        DoEvents
        ' Abbruch durch Benutzer prüfen!
        If bolCancel Then
            bolCancel = False
            If MsgBox("Prozedur wirklich abbrechen?", vbYesNo + vbQuestion) = vbYes Then
                With frmControl
                    .txtError.Text = "Abbruch durch Benutzer!"
                End With
            End If
        End If
    Loop
End Sub
```

```

        .cmdCancel.Enabled = False
        .cmdStart.Enabled = True
        Exit Sub
    End With
End If
End If
... weiterer Programmablauf
Loop
...
End Sub

```

Listing 7: VBA-Code-Schnipsel der Programmierlogik zum Abbrechen der laufenden Hauptprozedur

Zur Erinnerung, ausgelöst wird der Abbruch durch die Klick-Prozedur `cmdCancel_Click`. Dort wird der Abbruchschalter `bolCancel = True` gesetzt.

8 Hauptprozedur

Die Hauptprozedur steuert das Durchlaufen der ersten Tabelle (s. Tabelle 1) im Quelldokument mit der Schleife `Do While lngRow <= lngLastRow`, wobei die Variable `lngRow` den Schleifenzähler bzw. die jeweilige Tabellenzeile repräsentiert.

Die Hauptprozedur benötigt zwei untergeordnete VBA-Routinen:

Name der Routine	Typ der Routine	Aufgabe
fctHasPattern	Funktion	Mustersuche durchführen und Trefferzahl zurückgeben
HaveABreak	Unterprogramm	Schleifendurchlauf für bestimmte Dauer verzögern

Nach dem Durchlaufen der oben genannten *Do-While-Schleife* wird zur Sicherheit geprüft, ob die Zahl der Zeichen im Zieldokument vor und nach der Suche nach Mustern gleich geblieben ist, mit anderen Worten, dass die Funktion `fctHasPattern` keine ungeplanten Änderungen vorgenommen hat.

Unabhängig vom Prüfergebnis wird das Zieldokument zur Sicherheit *ohne* mögliche Änderungen geschlossen.

```

Public Sub ReadAndUpdateTable()
    ' Aufgabe: Externe Word-Tabelle durchlaufen und aktualisieren
    Dim bolRet As Boolean ' Rückgabewert
    Dim docSrc As Document ' Quelldokument
    Dim docTgt As Document ' Zieldokument
    Dim lngChar2 As Long ' Zeichenzähler nach dem Ende der Hauptprozedur
    Dim lngHits As Long ' Treffer bei der Mustersuche
    Dim lngRow As Long ' Zähler für Tabellenzeilen
    Dim lngLastRow As Long ' Gesamtzahl der Tabellenzeilen
    Dim rngcell As Range ' Tabellenzelle
    Dim strFind As String ' Suchbegriff
    Dim strTask As String ' Beschreibung des Suchmusters
    Dim strFileNm As String ' Name der Zieldatei
    Dim tbl As Table ' Word-Tabelle im Quelldokument (docSrc)
    On Error GoTo Err_Point
    Application.ScreenUpdating = False ' Bildschirmaktualisierung deaktivieren
    ' 1. Spalte der Word-Tabelle zurücksetzen
    Set docSrc = ActiveDocument
    Set tbl = docSrc.Tables(1) ' 1. Tabelle im aktuellen Quell-Dokument
    lngLastRow = tbl.Rows.Count - 1

```

```

For lngRow = 2 To lngLastRow
    With tbl.cell(lngRow, 1).Range
        .Font.Bold = False
    End With
Next lngRow
' Den vollständiger Name der Zieldatei aus dem Benutzerformular auslesen
strFileNm = frmControl.txtTgtNm
If Len(strFileNm) > 0 Then
    Set docTgt = Documents.Open(FileName:=strFileNm, _
        ReadOnly:=True, AddToRecentFiles:=False, Visible:=False)
Else
    MsgBox "Fehler bei der Auswahl der Zieldatei", vbCritical, "ReadAndUpdateTable"
    Exit Sub
End If
docSrc.Activate
' Laufbereich des Zeilenzählers vorgeben:
lngRow = 2 ' ab zweite Tabellenzeile
With frmControl
    .txtRow.Value = lngRow
    .txtRows.Value = lngLastRow ' bis vorletzte Tabellenzeile
End With
' Externe Word-Tabelle mit Do-While-Schleife durchlaufen
Do While lngRow <= lngLastRow
    DoEvents
    ' Abbruch durch Benutzer prüfen
    If bolCancel Then
        bolCancel = False
        If MsgBox("Prozedur wirklich abbrechen?", vbYesNo + vbQuestion) = vbYes Then
            With frmControl
                .txtError.Text = "Abbruch durch Benutzer!"
                .cmdCancel.Enabled = False
                .cmdStart.Enabled = True
            End With
            Exit Sub
        End With
    End If
End If
' Fehlermeldung im Benutzerformular zurücksetzen
With frmControl.txtError
    .Value = "keine"
    .BackColor = vbWhite
End With
' Inhalt der jeweiligen Zelle in Spalte 1 auslesen
Set rngcell = tbl.cell(lngRow, 1).Range
rngcell.End = rngcell.End - 1
strFind = rngcell.Text
frmControl.txtPattern = strFind
' Inhalt der jeweiligen Zelle in Spalte 3 auslesen
Set rngcell = tbl.cell(lngRow, 3).Range
rngcell.End = rngcell.End - 1
strTask = SplitCellContent(rngcell.Text, frmControl.chkEnglisch.Value)
frmControl.txtTask = strTask
' Rückkehrwert setzen
bolRetVal = False
' Änderungen nachverfolgen ausschalten
docTgt.TrackRevisions = False
' Suche nach Mustern im Zieldokument durchführen
lngHits = HasStrFind(docTgt, strFind)
' Ergebnis der Suche nach Mustern anzeigen
With frmControl.txtHits
    .Value = lngHits

```



```

        If .Value = 0 Then
            .BackColor = vbWhite
        Else
            .BackColor = vbRed
        End If
    End With
    ' 1. Spalte der externen Word-Tabelle aktualisieren
    With tbl.cell(lngRow, 1).Range
        .Font.Bold = bolRetVal ' Fettdruck
    End With
    ' Wartezeit ausführen
    With frmControl.txtWaitTm
        If .Value > 0 Then
            Call HaveABreak(CInt(frmControl.txtWaitTm.Value))
        End If
    End With
    ' Benutzerformular aktualisieren
    With frmControl
        .txtRow.Value = lngRow
        .Repaint
    End With
    ' Zeilenzähler der externen Word-Tabelle erhöhen
    lngRow = lngRow + 1
Loop ' Ende der Do-While-Schleife
' Zeichenzahl im Zieldokument nach Schleifendurchlauf ausgeben
lngChar2 = docTgt.Range.Characters.Count
With frmControl
    .txtCharCount2 = lngChar2
    .cmdStart.Enabled = False
    .cmdCancel.Enabled = False
    .cmdClose.Enabled = True
End With
Exit_Point:
On Error Resume Next
docTgt.Close SaveChanges:=wdDoNotSaveChanges
If docSrc.Saved = False Then docSrc.Save
Application.ScreenUpdating = True ' Bildschirmaktualisierung zulassen
Set docTgt = Nothing
Exit Sub
Err_Point:
MsgBox "Laufzeitfehler " & Err.Description, vbExclamation, "ReadAndUpdateTable"
Set tbl = Nothing
Set docSrc = Nothing
Resume Exit_Point
End Sub

```

Listing 8: Hauptprozedur

8.1 Suche mit Mustervergleich im Zieldokument durchführen

Die Funktion „*fctHasPattern*“ gibt die Zahl der Treffer zurück, wenn im Zieldokument eine Suche mit Mustervergleich unter Verwendung von Platzhaltern durchgeführt wird.

Außerdem wird mit dieser Funktion die logische Variable *bolRetVal* aktualisiert, die in allen Modulen bekannt ist, weil sie die Eigenschaft „*public*“ besitzt. Wenn `bolRetVal = True` gesetzt wird, ändert die aufrufende Prozedur mit dem Namen „*ReadAndUpdateTable*“ (s. Listing 8: Hauptprozedur

) den entsprechenden Eintrag in der Spalte „Suchen nach“ der externen Tabelle im Quelldokument in **Fettdruck**. Folglich kann nach dem normalen Ende der Hauptprozedur aus Tabelle 1 entnommen werden, bei welchem Eintrag in der Spalte „Suchen nach“ 1 oder mehr Treffer vorgekommen sind.

```

Private Function fctHasPattern(ByRef docTgt As Document, _
    ByVal strFind As String, ByVal bolWildcards As Boolean) As Long
    ' Aufgabe: Suchbegriff strFind im Zieldokument finden und Trefferzahl zurückgeben
    ' Argumente: docTgt: Zieldokument
    ' strFind: Suchmuster
    ' bolWildcards: Platzhalter verwenden
    Dim lngLoop As Long ' Schleifenzähler
    Dim lngHits As Long ' Trefferzahl
    Dim rngTgt As Range ' Suchbereich im Zieldokument
    On Error GoTo Err_Point
    Set rngTgt = docTgt.Range(0, 0)
    With rngTgt.Find
        .ClearFormatting
        .Text = strFind
        .Forward = True
        .Format = False
        .Wrap = wdFindStop
        .MatchCase = False
        .MatchWholeWord = False
        .MatchByte = False
        .MatchAllWordForms = False
        .MatchSoundsLike = False
        .MatchWildcards = bolWildcards ' Platzhalter verwenden
    Do While .Execute
        bolRetVal = .Found ' ... ist eine Public-Variable!
        lngHits = lngHits + 1 ' Trefferzahl erhöhen
        lngLoop = lngLoop + 1 ' Schleifenzähler erhöhen
        If lngLoop > 100 Then ' Endlosschleife verhindern
            With frmControl.txtError
                .Value = "Endlosschleife: " & strFind
                .BackColor = vbRed
            End With
            lngHits = 0
            Exit Do
        End If
        rngTgt.Collapse wdCollapseEnd
        DoEvents
    Loop
    End With
Exit_Point:
    fctHasPattern = lngHits ' Rückgabewert: Trefferzahl
    Exit Function
Err_Point:
    Dim strErrMsg As String
    If Err.Number = 5560 Then ' Syntaxfehler im Feld 'Suchen nach'
        strErrMsg = "Syntaxfehler: " & strFind
    Else
        strErrMsg = "Laufzeitfehler: " & Err.Description
    End If
    With frmControl.txtError
        .Value = strErrMsg
        .BackColor = vbRed
    End With
    lngHits = 0
    Resume Exit_Point

```

```
End Function
```

Listing 9: Funktion zur Mustersuche mit 3 Argumenten und 1 Rückgabewert

8.2 Ablauf der Hauptprozedur verzögern

Die Subprozedur „HaveABreak“ verzögert den Ablauf der Hauptprozedur für bestimmte Dauer, damit die jeweils aktualisierte Anzeige des Formulars „frmControl“ (s. Abbildung 2) vom Benutzer kurz gelesen werden kann.

```
Public Sub HaveABreak(intSeconds As Integer)
    ' Aufgabe: Ablauf der Hauptprozedur für bestimmte Dauer (Sekunden) anhalten
    Dim lngPauseTime As Long
    Dim lngStart As Long
    lngPauseTime = intSeconds
    lngStart = Timer
    Do While Timer < lngStart + lngPauseTime
        DoEvents
    Loop
End Sub
```

Listing 10: Ablauf der Hauptprozedur für bestimmte Dauer anhalten

9 E-Mail-Link

Das Benutzerformular (s. Abbildung 2) enthält unten links einem Verweis auf die E-Mail-Adresse des Autors:

```
Private Sub lblHyperlink_Click()
    ActiveDocument.FollowHyperlink Address:="mailto:volker@dr-thormaehlen.de", NewWindow:=True
End Sub
```

Listing 11: Hyperlink zur E-Mail-Adresse des Autors

10 Laufzeiten

Die gemessenen Laufzeiten⁷ der Hauptprozedur angewandt auf ein Zieldokument mit 105 bzw. 329 Seiten (bei 2 Sekunden Verzögerung je Suchmuster) enthält Tabelle 3:

Zieldokument mit ...	Laufzeit im Format hh:mm:ss ...
105 Seiten, 301.193 Zeichen	00:01:14
329 Seiten, 1.124.129 Zeichen	00:01:20

Tabelle 3: Experimentelle Laufzeiten der Hauptprozedur bei 2 Sekunden Wartezeit

Ohne Wartezeit ergibt sich (s. Tabelle 4):

Zieldokument mit ...	Laufzeit im Format hh:mm:ss ...
105 Seiten, 301.193 Zeichen	00:00:17
329 Seiten, 1.124.129 Zeichen	00:00:26

Tabelle 4: Experimentelle Laufzeiten der Hauptprozedur ohne Wartezeit

⁷ Application.Version = 14, Intel-Prozessor mit 3,13 GHz

11 Grenzen

Die *Erweiterte Suche mit Platzhaltern* findet ihre Grenzen, wenn beispielsweise beim Korrekturlesen und/oder der formalen Bereinigung umfangreicher Word-Dokumente (z. B. wissenschaftliche Arbeiten, Gutachten, Lebenserinnerungen und dergleichen), nach unpaarigen Klammern⁸, unpaarigen Anführungszeichen⁹ oder Ähnlichem gesucht wird, egal, ob global, je Abschnitt oder je Absatz. Meistens ist dann der Einsatz von speziellen VBA-Prozeduren angezeigt. Vgl. dazu [3] und die dort angegebene Literatur sowie [4].

12 Anhang

Die Einträge in der Spalte „Suchen nach“ (s. Tabelle 1) können bei Bedarf in entsprechende Bereinigungsverfahren eingebaut werden. Dies wird am Beispiel des Eintrags ([!^13^0147.;;\?!\!])(^13) demonstriert:

```
Sub InsertMissingDotBeforeParagraphBreak(docTgt As Document)
' Aufgabe: Fehlenden Punkt vor einer Absatzendemarke einfügen
'           docTgt: Zieldokument
Dim rng      As Range ' Suchbereich
Dim lngHits  As Long  ' Zahl der Treffer
Dim lngInsCnt As Long  ' Zahl der Ersetzungen
Set rng = docTgt.Range(0, 0)
With rng.Find
    .ClearFormatting
    .Text = "[!^13^0147.;;\?!\!](^13)"
    .Forward = True
    .Wrap = wdFindStop
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchByte = False
    .MatchAllWordForms = False
    .MatchSoundsLike = False
    .MatchWildcards = True
    .Execute
Do While .Found
    With rng
        .Select
        lngHits = lngHits + 1
        If vbYes = MsgBox("Fehlenden Punkt vor Absatzendezeichen einfügen?", _
            vbYesNo + vbQuestion, "Ersetzen?") Then
            .Text = Left(rng.Text, 1) & "." & Right(rng.Text, 1)
            lngInsCnt = lngInsCnt + 1
        End If
        .Collapse Direction:=wdCollapseEnd
    End With
    .Execute
Loop
MsgBox CStr(lngHits) & " Fundstellen, davon " & _
    CStr(lngInsCnt) & " berichtigt.", vbInformation, "Ergebnis:"
End With
End Sub
```

Listing 12: Beispiel einer interaktiven Prozedur zum Suchen und Ersetzen mit Platzhaltern in einem Zieldokument

⁸ Gemeint sind runde, eckige, spitze und/oder geschweifte Klammern.

⁹ Dazu zählen doppelte, halbe, typografische und/oder spitze Anführungszeichen (sog. Guillemets o. Chevrons).

Bevor die Routine (s. Listing 12) aufgerufen wird, sollten zur Sicherheit folgende 6 Einstellungen mittels VBA-Code vorgenommen werden:

- Änderungen nachverfolgen ausschalten
- Änderungen und Kommentare ausblenden
- Bildschirmaktualisierung ausschalten
- Dokumentschutz aufheben
- Feldcodes ausblenden
- Versteckten Text ausblenden

13 Abbildungsverzeichnis

Abbildung 1: Suchen und Ersetzen mit Platzhaltern.....	3
Abbildung 2: Layout des Benutzerformulars (Name: frmControl).....	9

14 Tabellenverzeichnis

Tabelle 1: Einträge für die sog. Erweiterte Suche in einem Word-Dokument	8
Tabelle 2: Steuerelemente des Benutzerformulars	10
Tabelle 3: Experimentelle Laufzeiten der Hauptprozedur bei 2 Sekunden Wartezeit	19
Tabelle 4: Experimentelle Laufzeiten der Hauptprozedur ohne Wartezeit	19

15 Quellcodelistings

Listing 1: Benutzerformular ungebunden aufrufen	8
Listing 2: Klick-Prozedur zur Auswahl des Zieldokuments und zur anfänglichen Belegung der Felder des Benutzerformulars	12
Listing 3: Funktion zur Auswahl einer Word-Datei im Dialog	12
Listing 4: Klick-Prozedur starten und Hauptprozedur aufrufen	13
Listing 5: Klick-Prozedur zum Abbrechen der Hauptprozedur	14
Listing 6: Klick-Prozedur zum Schließen des Benutzerformulars	14
Listing 7: VBA-Code-Schnipsel der Programmierlogik zum Abbrechen der laufenden Hauptprozedur	15
Listing 8: Hauptprozedur	17
Listing 9: Funktion zur Mustersuche mit 3 Argumenten und 1 Rückgabewert.....	19
Listing 10: Ablauf der Hauptprozedur für bestimmte Dauer anhalten	19
Listing 11: Hyperlink zur E-Mail-Adresse des Autors	19
Listing 12: Beispiel einer interaktiven Prozedur zum Suchen und Ersetzen mit Platzhaltern in einem Zieldokument	20

Literaturverzeichnis

- [1] G. Mayor und K. Linke, „Finding and replacing characters using wildcards,“ [Online]. Available: <http://word.mvps.org/faqs/general/usingwildcards.htm>. [Zugriff am 11 08 2016].
- [2] Microsoft, „Find and replace text and other data in a Word document,“ [Online]. Available: <https://support.office.com/en-us/article/Find-and-replace-text-and-other-data-in-a-Word-document-c6728c16-469e-43cd-afe4-7708c6c779b7?CorrelationId=f9e589d9-7b74-4f1a-8704-f80e7300b6be&ui=en-US&rs=en-US&ad=US&ocmsassetID=HA102350661>. [Zugriff am 09 05 2017].
- [3] V. Thormählen, „Word-Dokument säubern (CleanUp_Document),“ 10 11 2016. [Online]. Available: http://dr-thormaehlen.de/WORD/CleanUp_Document.pdf. [Zugriff am 16 05 2017].
- [4] L. Harnby, „The author’s proofreading companion,“ 11 02 2017. [Online]. Available: https://www.louiseharnbyproofreader.com/uploads/9/5/8/7/9587442/author_proofreading_companion.pdf. [Zugriff am 06 05 2017].

