
Abkürzungsverzeichnis für ein Word-Dokument automatisch erstellen:

Akronyme tabellieren

Dr. Volker Thormählen, Ratingen

28.06.2016

Zur Erstellung eines Abkürzungsverzeichnisses besitzt das Textprogramm MS Word keine Standard-Funktionalität wie bei Abbildungs- und Tabellenverzeichnissen. In diesem Beitrag wird deshalb beschrieben, wie die in einem Word-Dokument gefundenen Akronyme im entsprechenden Abkürzungsverzeichnis mit Hilfe von VBA-Programmen automatisch tabelliert werden können. Dieser Vorgang kann beliebig oft wiederholt werden, was bei nachträglichen Änderungen eines umfangreichen Word-Dokuments, die sich auf die Seitennummerierung auswirken, vorteilhaft ist.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Tabellenverzeichnis.....	II
Abbildungsverzeichnis.....	III
Abkürzungsverzeichnis.....	IV
1 Aufgabenstellung	1
2 Voraussetzungen	2
3 Word-Dokumente.....	3
4 Tabellenaufbau.....	4
5 Programmorganisation.....	5
5.1 Abkürzungsverzeichnis finden	6
5.2 Tabelle einfügen.....	6
5.3 Suchbereich bestimmen	6
5.4 Suchbereich durchlaufen	6
5.5 Suchergebnisse ausgeben.....	7
5.6 Umgebenden Text finden	7
6 Funktionstest.....	8
7 Dateiauswahl.....	9
8 Zusammenfassung.....	10
Listings der Programme.....	V
Literatur-/Quellenverzeichnis.....	XIII

Tabellenverzeichnis

Listing 1: Das Wort <i>Abkürzungsverzeichnis</i> im Zieldokument finden	V
Listing 2: Gefundene Akronyme tabellieren, Teil 1 von 3	VI
Listing 3: Gefundene Akronyme tabellieren, Teil 2 von 3	VII
Listing 4: Gefundene Akronyme tabellieren, Teil 3 von 3	VIII
Listing 5: Bedeutung der Akronyme bei bestimmter Schreibweise im Text ermitteln....	IX
Listing 6: Alte Tabelle der Akronyme löschen und zugehörige Textmarke erneuern.....	X
Listing 7: Neue Tabelle der Akronyme formatieren.....	X
Listing 8: Auswahl der Zieldatei mit eingebautem Word-Dialog	XI
Listing 9: Ersatz f. Listing 1 bei Auswahl d. Zieldatei mit eingebautem Word-Dialog.....	XII

Abbildungsverzeichnis

Abbildung 1: Zieldokument(e) und Quelldokument	3
Abbildung 2: Programmorganisation	5

Abkürzungsverzeichnis

Akronym	Bedeutung	Seite
AUDI		8
BMW	Bayerische Motorenwerke	8
VBA	Visual Basic for Applications	1
VW	Volkswagen	8

1 Aufgabenstellung

In diesem Beitrag wird beschrieben, wie alle Akronyme¹ in einem Word-Dokument gefunden und als geordnete Tabelle in das dazugehörige Abkürzungsverzeichnis eingefügt werden können. Der Suchvorgang erfolgt durch Programme, die mit der Programmiersprache VBA (Visual Basic for Applications) erstellt wurden. Im Folgenden wird im Detail beschrieben,

- welche Voraussetzungen erfüllt sein müssen, um den Suchvorgang zu automatisieren,
- welche Spalten die Tabelle der Akronyme umfasst,
- welche Programme bzw. Makros² für die Tabellierung der Akronyme notwendig sind,
- wie diese Programme organisiert sind,
- wie sie getestet und eingesetzt werden können.

Der Beitrag endet mit einer kurzen Zusammenfassung und den Listings der eingesetzten Programme.

¹ Ein Akronym ist ein Kurzwort, das aus den Anfangsbuchstaben mehrerer Wörter zusammengesetzt ist.

² Ein Makro ist ein kleines Programm, das bestimmte Abläufe in einem Anwendungsprogramm automatisiert.

2 Voraussetzungen

Das Word-Dokument, das die zu tabellierenden Akronyme enthält, muss folgende Eigenschaften besitzen:

- Es muss in logische Abschnitte eingeteilt sein, wie z. B. bei wissenschaftlichen Arbeiten üblich³,
- Jeder logische Abschnitt muss mit einen *Abschnittsumbruch (Neue Seite)* enden,
- Das Wort „Abkürzungsverzeichnis“ muss im Dokument vorkommen und mit der benutzerdefinierten Formatvorlage „ÜberschriftohneNr“ verbunden sein.
- Akronyme müssen im Text als zwei oder mehr aufeinander folgende Großbuchstaben vorkommen.

Optional kann die Bedeutung/Definition eines Akronyms im Text nach folgenden syntaktischen Regeln aufgenommen werden:

- Die Bedeutung steht in runden Klammern **vor** dem Akronym, also: **(Bedeutung) Akronym**
- Die Bedeutung folgt in runden Klammern **nach** dem Akronym, also: **Akronym (Bedeutung)**

Beide Schreibweisen können im selben Dokument vorkommen.

³ Vgl. dazu die 8-teilige Video-Serie von (Puls, Erika, 2013).

3 Word-Dokumente

Das Word-Dokument, das die Akronyme enthält, wird im Folgenden *Zieldokument* genannt. Die entsprechende Word-Datei besitzt gewöhnlich den Zusatz **docx** zum Dateinamen.

Das Word-Dokument, das die Programme (Makros) enthält, wird im Folgenden *Quelldokument* genannt. Die entsprechende Word-Datei besitzt gewöhnlich den Zusatz **docm** zum Dateinamen. Der Name des Zieldokuments ist im Hauptprogramm des Quelldokuments als Konstante hinterlegt. Diese Konstante (**conTgtNm**) muss ggf. angepasst werden, siehe Listing 1. Sonst entsteht ein Laufzeitfehler.

Die Programme im *Quelldokument* können (nach Anpassung der Konstante **conTgtNm**) immer wieder auf verschiedene *Zieldokumente* angewendet werden, aber sinnvollerweise nur dann, wenn die oben genannten syntaktischen Voraussetzungen erfüllt sind.

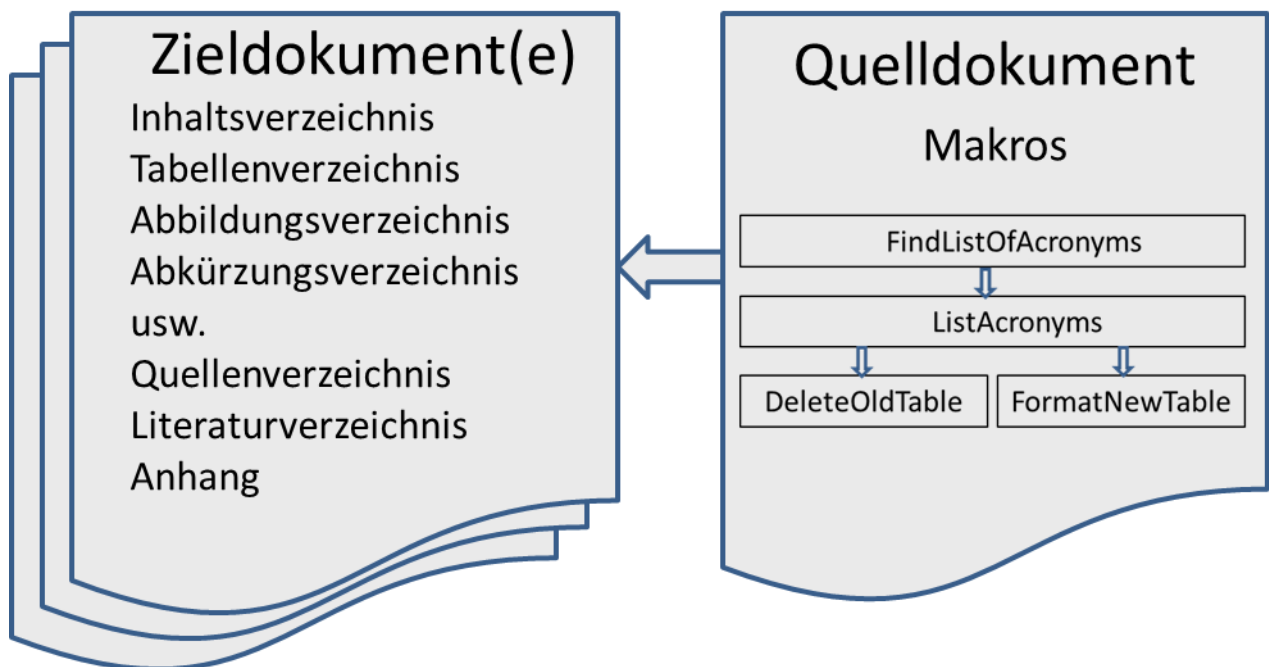


Abbildung 1: Zieldokument(e) und Quelldokument

4 Tabellenaufbau

Die Tabelle der Akronyme besteht aus 3 Spalten:

- Akronym
- Bedeutung eines Akronyms
- Seitennummer des ersten Vorkommens eines Akronyms

Wenn ein Akronym beim automatisierten Suchen im Dokument gefunden wird, werden die erste und dritte Spalte mit den entsprechenden Angaben gefüllt. Wenn zu einem Akronym eine vor- oder nachgestellte Bedeutung bzw. Definition gehört, wird diese ebenfalls automatisch gefunden und jeweils in die zweite Tabellenspalte eingesetzt. Sonst bleibt die zweite Tabellenspalte leer und muss ggf. später manuell ergänzt werden.

Wenn mehr als 1 Akronym in Zieldokument gefunden wird, wird die Tabelle nach dem Inhalt in der ersten Spalte aufsteigend sortiert.

Die Spaltenköpfe in der Kopfzeile der Tabelle werden fett formatiert und ihr Hintergrund hellgrau schattiert.

5 Programmorganisation

Zum Finden und Tabellieren von Akronymen in einem Zieldokument werden diese 4 Makros benutzt (siehe Abbildung 2):

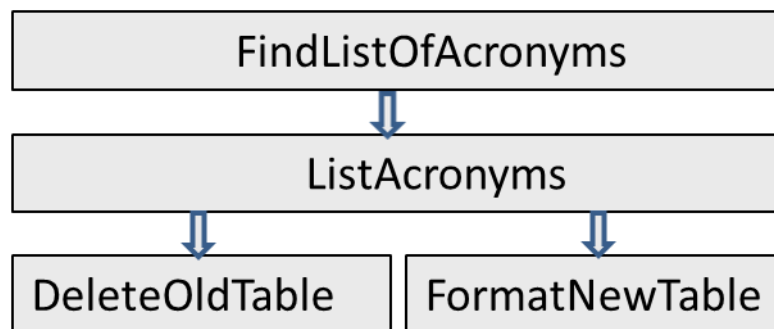


Abbildung 2: Programmorganisation

Die Hauptaufgabe von **FindListOfAcronyms** ist, im Zieldokument die Überschrift *Abkürzungsverzeichnis* zu finden und das Unterprogramm **ListAcronyms** aufzurufen.

Das Unterprogramm **ListAcronyms** sucht im Zieldokument nach Akronymen in einem Abschnitt (engl. *Section*), dessen erster Absatz (engl. *Paragraph*) mit der eingebauten Formatvorlage *Überschrift 1* verbunden ist. Es wird dabei angenommen, dass nur der zugehörige Abschnitt nach Akronymen durchsucht werden soll, also *nicht* das ganze Zieldokument. Die beiden untergeordneten Programme (siehe Abbildung 2) werden von dort aus automatisch aufgerufen und ausgeführt.

Die Unterprogramme **DeleteOldTable** und **FormatNewTable** dienen im Wesentlichen dazu, die Wiederholbarkeit der Anwendung sicherzustellen. Diese Eigenschaft ist wichtig, wenn im Zieldokument nachträglich Änderungen vorgenommen werden, die zu einer Verschiebung der Seitenzahlen führen. Dann müssen die Seitenzahlen der Akronyme aktualisiert werden.

Wie erwähnt, befinden sich die Programme als sog. *Makros* in einem beliebig benannten Quelldokument mit dem Namenszusatz ***.docm**. Es enthält gewöhnlich keinen Fließtext, sondern nur die vier genannten Makros. Wie diese in einem Quelldokument installiert werden können, wird sehr anschaulich beschrieben von Graham Mayor (Mayor, 2012). Die erforderlichen Installationsschritte werden deshalb hier nicht weiter beschrieben.

Das Word-Dokument, in dem sich die gesuchten Akronyme befinden und wo sie in einer Tabelle unter der Überschrift „*Abkürzungsverzeichnis*“ ausgewiesen werden sollen, wird, wie gesagt, Zieldokument genannt. Der Namenszusatz ist gewöhnlich ***.docx**. Der Name des Zieldokuments wird im ersten Unterprogramm als Konstante (**conTgtNm**) hinterlegt.

5.1 Abkürzungsverzeichnis finden

Hauptaufgabe des Programms *FindListOfAcronyms* ist, genau die Stelle im Zieldokument zu finden, wo die Tabelle der Akronyme eingefügt werden soll. Nur wenn diese Stelle gefunden wird, kann dort die Tabelle der Akronyme erstellt werden.

5.2 Tabelle einfügen

Das Unterprogramm *ListAcronyms* wird vom Makro *FindListOfAcronyms* aufgerufen und sorgt dafür, dass die Tabelle der Akronyme an vorgegebener Stelle im Zieldokument eingesetzt wird, nämlich bei der Textmarke **TabSpot** unterhalb der Überschrift „Abkürzungsverzeichnis“.

Zunächst wird die möglicherweise bereits vorhandene Tabelle der Akronyme gelöscht, um Platz für eine neue, aktuelle Tabelle zu schaffen. Dadurch ist es möglich, den gesamten Suchvorgang beliebig oft zu wiederholen. Das ist vorteilhaft bei umfangreichen Dokumenten, wenn sich die Seitennummern durch nachträgliche Änderungen verschieben.

5.3 Suchbereich bestimmen

Das Programm *ListAcronyms* bestimmt u. a., in welchem Abschnitt des Zieldokuments nach Akronymen gesucht werden soll. Dieser Suchbereich wird festgelegt, indem nach dem ersten Vorkommen eines Absatzes gesucht wird, der mit der eingebauten Formatvorlage „Überschrift 1“ formatiert ist. Wegen der vorausgesetzten Einteilung des Zieldokuments nach logischen Abschnitten beginnt genau dort der Abschnitt (engl. *Section*), der durchsucht werden soll. Damit wird erreicht, dass nicht relevante Abschnitte bei der Suche unberücksichtigt bleiben.

5.4 Suchbereich durchlaufen

Mit einer *Do-Until-Schleife* wird im zuvor festgelegten Suchbereich nach Akronymen gesucht. Das dafür verwendete Suchmuster, gespeichert in *strPattern*, besteht aus einem sog. *regulären Ausdruck*⁴:

```
strPattern = "<[A-Z]{2" & strListSep & ">"
```

⁴ Mit einem *regulären Ausdruck* (engl. regular expression, kurz: regex oder regexp) werden Zeichenketten auf bestimmte Eigenschaften bzw. Muster geprüft.

Dieses Suchmuster bedeutet von links nach rechts:

- < Beginn des gesuchten Wortes
- [A-Z] Zeichenauswahl in eckigen Klammern: Hier nur Großbuchstaben von A bis Z
- { Beginn der Wiederholungsfaktoren, sog. Quantoren
- 2 zwei oder mehr Zeichen aus der Zeichenauswahl [A-Z]
- & Operator zur Verkettung von Zeichenketten
- `strListSep` Die Variable `strListSep` enthält das jeweils gültige Trennzeichen⁵ für Listen
- & Operator zur Verkettung von Zeichenketten
- } Ende der Wiederholungsfaktoren
- > Ende des gesuchten Wortes

Mit dem Suchmuster in `strPattern` werden Zeichenketten mit 2 oder mehr Großbuchstaben in Folge als gesuchte Akronyme identifiziert. Zusammen mit der jeweils dazu gehörigen Seitennummer werden diese beiden Informationen zwischengespeichert, um daraus in einem späteren Schritt die gewünschte Tabelle der Akronyme zu erzeugen.

5.5 Suchergebnisse ausgeben

Die zwischengespeicherten Informationen werden an der gewünschten Stelle hinter der Überschrift „*Abkürzungsverzeichnis*“ gefügt. In einem nächsten Schritt werden sie in eine 3-spaltige Tabelle umgewandelt⁶. Wird mehr als 1 Akronym gefunden, wird die Tabelle nach dem Inhalt in der 1. Spalte aufsteigend sortiert. In einem letzten Schritt wird das Aussehen der Tabelle verschönert. Dazu dient das Programm mit dem Namen `FormatNewTable`.

5.6 Umgebenden Text finden

Wenn im definierten Suchbereich nicht nur Akronyme gefunden werden sollen, sondern auch die jeweils dazugehörige Bedeutung bzw. Definition, müssen die Codezeilen aus Listing 5 an der grau hinterlegten Stelle in Listing 3 eingefügt werden. Lediglich aus didaktischen Gründen werden diese Codezeilen in Listing 5 separat dargestellt. Es ist aber praktischer, auf die Aussonderung zu verzichten.

Je nach Schreibweise der gefundenen Akronyme mit vor- oder nachgestelltem Text wird rechts bzw. links davon nach der Bedeutung/Definition gesucht, siehe Listing 5

⁵ Je nach Land entweder ein Komma oder ein Semikolon (Letzteres für Deutschland)

⁶ Das geht erheblich schneller, als wenn eine bestehende Tabelle Zeile für Zeile und Zelle für Zelle belegt wird.

Funktionstest

Die Eignung der vorgestellten Unterprogramme zur Erstellung einer 3-spaltigen Tabelle mit Akronymen nach der Überschrift „*Abkürzungsverzeichnis*“ soll mit 3 Kurzwörtern aus dem Bereich der Automarken demonstriert werden:

- VW (Volkswagen)
- (Bayerische Motorenwerke) BMW
- AUDI

Das Ergebnis des Funktionstests ist im vorliegenden Dokument unter der Überschrift „*Abkürzungsverzeichnis*“ zu sehen. Erwartet werden 3 Einträge für die soeben erwähnten Kurzwörter sowie 1 weiterer Einträge für das Kürzel VBA, das in der *Aufgabenstellung* (siehe Seite 1) vorkommt.

Die Bedeutung/Definition des Kurzworts AUDI kommt im Suchbereich *nicht* vor und folglich auch nicht in der Tabelle. Der fehlende Eintrag in der zweiten Tabellenspalte muss daher manuell eingetragen werden.

Die Seitennummern in der dritten Spalte der Tabelle geben nur das erste Vorkommen eines Akronyms an. Jedes weitere Vorkommen wird ignoriert, vgl. dazu in Listing 3 folgende If-Abfrage:

```
If InStr(strOutput, strAcronym) = 0 Then ... End If.
```

Die Textfunktion *Instr* gibt 0 zurück, wenn *strAcronym* innerhalb von *strOutput* nicht vorkommt.

6 Dateiauswahldialog

Die Auswahl des Zieldokuments kann durch den eingebauten Word-Dialog *Datei öffnen* variabel gestaltet werden. Dafür kann die Funktion *OpenTargetFile* eingesetzt werden (siehe Listing 8). Allerdings muss dann Listing 1 durch Listing 9 ersetzt werden. Dort ist der Aufruf der Funktion *OpenTargetFile* grau markiert.

7 Zusammenfassung

Die Besonderheiten des beschriebenen Ansatzes zum Tabellieren von Akronymen im Abkürzungsverzeichnis eines Zieldokuments sind:

- Einsatz strukturierter, ausführlich kommentierter VBA-Programme, die sich in einem beliebig benannten Quelldokument befinden.
- Möglichkeit zur Wiederverwendung dieser Programme für beliebig viele Zieldokumente, die jeweils den beschriebenen *Voraussetzungen* entsprechen.
- Berücksichtigung der Bedeutung bzw. Definition von Akronymen bei syntaktisch richtiger Schreibweise im Text des Zieldokuments (siehe Listing 5)
- Eingrenzung der Suche nach Akronymen auf einen genau bestimmbaren Abschnitt des Zieldokuments (siehe Listing 2)
- Wiederholbarkeit der Suche, was bei nachträglichen Änderungen im Zieldokument vorteilhaft ist, weil die Seitenzahl der jeweiligen Fundstelle in der Tabelle der Akronyme automatisch aktualisiert wird.

Listings der Programme

Die Listings der Programme enthalten viele Kommentare, damit sie auch von Personen verstanden werden können, die sich mit dem Schreiben von VBA-Makros nicht auskennen. Die Kommentare dienen im Wesentlichen dazu, die Abläufe im jeweiligen Programm zu verdeutlichen.

```
Sub FindListOfAcronyms ()
    Const conTgtNm      As String = "Zieldokument" ' Name des Zieldokuments: Bitte abpassen
    Const conExt       As String = ".docx"        ' Dateizusatz
    Dim docSrc         As Document                ' Quelldokument
    Dim docTgt         As Document                ' Zieldokument
    Dim strTgtPath     As String                  ' Pfad zum Zieldokument
    Dim strTgtFullNm   As String                  ' voller Name des Zieldokuments
    Dim lngSctn       As Long                     ' Nr. des relevanten Abschnitts im
Zieldokument
    Application.ScreenUpdating = False           ' Bildschirmaktualisierung ausschalten
    On Error GoTo ErrHandler                     ' Fehlerbehandlungsroutine
    Set docSrc = ActiveDocument                  ' Verweis zum Quelldokument setzen
    strTgtPath = docSrc.Path                     ' Pfad zum Zieldokument bestimmen
    If Right(strTgtPath, 1) <> "\" Then strTgtPath = strTgtPath + "\"
    strTgtFullNm = strTgtPath & conTgtNm & conExt
    If Dir(strTgtFullNm) = vbNullString Then
        MsgBox "Word-Datei '" & strTgtFullNm & "' nicht gefunden!", vbCritical
        GoTo ExitHandler
    End If
    Documents.Open FileName:=strTgtFullNm       ' Zieldatei öffnen
    Set docTgt = ActiveDocument                  ' Verweis zum Zieldokument setzen
    docTgt.Select                               ' Zieldokument auswählen
    With Selection.Find                         ' Überschrift 'Abkürzungsverzeichnis' finden
        .ClearFormatting
        .Forward = True
        .Format = True
        .Style = "ÜberschriftOhneNr"           ' benutzerdefinierte Formatvorlage
        .MatchCase = True
        .MatchWholeWord = True
        .Wrap = wdFindStop
        .Execute FindText:="Abkürzungsverzeichnis"
        If .Found = True Then
            Selection.Collapse wdCollapseEnd    ' Wenn Suche erfolgreich, dann ...
            Call ListAcronyms(docTgt)           ' Ergebnis der Suche auf den Endpunkt reduzieren
                                                ' Tabelle der Akronyme im Zieldokument erstellen
        End If
    End With
ExitHandler:
    On Error Resume Next
    If docTgt.Saved = False Then docTgt.Save    ' Zieldokument speichern, aber nicht schließen
    Set docTgt = Nothing                       ' Verbindung zum Zieldokument aufheben
    Application.VBE.MainWindow.Visible = False ' VBA-Editor ausblenden
    docSrc.Close SaveChanges:=wdSaveChanges    ' Quelldokument schliessen und speichern
    Set docSrc = Nothing                       ' Verbindung zum Quelldokument aufheben
    Application.ScreenUpdating = True          ' Bildschirmaktualisierung einschalten
    Exit Sub                                    ' Programm verlassen
ErrHandler:
    MsgBox Err.Description, vbCritical          ' Fehlermeldung anzeigen
    Resume ExitHandler                          ' Programm bei der Zeilenmarke "ExitHandler" fortsetzen
End Sub
```

Listing 1: Das Wort *Abkürzungsverzeichnis* im Zieldokument finden

Die benutzerdefinierte Formatvorlage *ÜberschriftOhneNr* legt u.a. fest, dass ein *Seitenumbruch oberhalb* des *Abkürzungsverzeichnisses* vorgenommen wird. Im Programm (siehe Listing 1) wird die Suche nach dem Wort *Abkürzungsverzeichnis* dadurch eingeschränkt.


```

Sub ListAcronyms(docTgt As Document)
' Die Konstante conBkmNm enthält den Namen eine Textmarke, die bestimmt,
' wo die Tabelle der Akronyme eingefügt werden soll.
Const conBkmNm As String = "TabSpot"
Dim strAcronym As String ' Akronym (d. h. Kurzwort, Kürzel)
Dim strOutput As String ' Zeichenkette als vorläufige Tabellenzeile
Dim strExplain As String ' Erklärung/Bedeutung eines Akronyms
Dim strListSep As String ' Trennzeichen Listen
Dim strPattern As String ' Suchmuster (regulärer Ausdruck)
Dim newTbl As Table ' neue Tabelle für gefundene Akronyme
Dim rngTbl As Range ' Tabellenbereich
Dim rngSctn As Range ' Suchbereich für Akronyme
Dim rngResult As Range ' verkleinerter Suchbereich für Akronyme
Dim rngPara As Range ' Absatzbereich
Dim para As Paragraph ' aktueller Absatz
Dim lngSctn As Long ' Abschnittsnummer
Dim lngPgNbr As Long ' Seitennummer eines Akronyms
Dim intCnt As Integer ' Zähler für gefundene Akronyme
Dim bolEmpty As Boolean ' Indikator für leere Absätze
On Error GoTo ErrHandler ' Fehleroutine definieren
With Application
.ScreenUpdating = False ' Bildschirmaktualisierung ausschalten
strListSep = .International(wdListSeparator) ' Trennzeichn f. Listen ermitteln
End With
' Anzeige von ausgeblendetem Text verhindern
ActiveWindow.View.ShowHiddenText = False
' Alte Tabelle der Akronyme löschen, falls die zugehörige Textmarke vorhanden is
Call DeleteOldTable(docTgt, conBkmNm)
' Ersten Absatz mit der Formatvorlage 'Überschrift 1' im Zieldokument finden
For Each para In docTgt.Paragraphs
If para.Style.NameLocal = "Überschrift 1" Then
Set rngPara = para.Range
rngPara.Select
' relevante Abschnittsnummer im Zieldokument bestimmen
lngSctn = rngPara.Information(wdActiveEndSectionNumber)
Exit For
End If
Next para
If lngSctn > 0 Then ' Wenn die relevante Abschnittsnummer im Zieldokument gefunden wurde, ...
' Such- und Ergebnisbereich für Akronyme im Zieldokument festlegen
Set rngSctn = docTgt.Sections(lngSctn).Range
Set rngResult = rngSctn.Duplicate
Else
MsgBox "Relevanten Abschnitt im Zieldokument für Akronyme nicht gefunden", vbCritical
GoTo ExitPoint ' Programm abbrechen
End If

```

Listing 2: Gefundene Akronyme tabellieren, Teil 1 von 3

Die grau markierte *For-Next*-Schleife in Listing 2 dient dazu, genau den Abschnitt (engl. *Section*) im Zieldokument zu bestimmen, in dem nach Akronymen gesucht werden soll.

```

' Suchmuster (regulärer Ausdruck) definieren: Findet 2 oder mehr Großbuchstaben in Folge
strPattern = "<[A-Z]{2" & strListSep & ">"
' Kopfzeile der (neuen) Tabelle für Akronyme definieren
strOutput = "Akronym" & vbTab & "Bedeutung" & vbTab & "Seite" & vbCr
Do ' fußgesteuerte Suchschleife
  With rngResult.Find
    .ClearFormatting
    .Text = strPattern ' Suchmuster für Akronyme in strPattern
    .Replacement.Text = vbNullString
    .Forward = True
    .Wrap = wdFindStop
    .Format = False
    .MatchCase = True
    .MatchWildcards = True ' Platzhaltersuche
    .MatchWholeWord = True
    .Execute
  End With
  If Not rngResult.Find.Found Then Exit Do
  rngResult.Select
  With Selection
    strAcronym = .Text ' Akronym
    lngPgNbr = .Information(wdActiveEndAdjustedPageNumber) ' Seitennummer
  End With
  '---Codezeilen aus Listing 5 ggf. hier einfügen bei best. Schreibweise der Akronyme
  ' Wenn das gefundene Akronym noch nicht im Output-String erhalten ist, dann ...
  If InStr(strOutput, strAcronym) = 0 Then
    intCnt = intCnt + 1 ' Zähler für Akronyme erhöhen
    ' Eine weitere neue Tabellenzeile vorbereiten
    strOutput = strOutput & strAcronym & _
      vbTab & strExplain & _
      vbTab & Str(lngPgNbr) & vbCr
  End If
  ' Suchbereich im relevanten Abschnitt verkleinern
  rngResult.MoveStart unit:=wdCharacter, Count:=Len(strAcronym)
  rngResult.End = rngSctn.End
Loop Until Not rngResult.Find.Found

```

Listing 3: Gefundene Akronyme tabellieren, Teil 2 von 3

Die grau markierte *Do-Until*-Schleife (siehe Listing 3) dient dazu, im zuvor ausgewählten Abschnitt des Zieldokuments nach Akronymen zu suchen und diese in der Zeichenkette *strOutput* zu sammeln.

Die grau markierte Kommentarzeile im Schleifenkörper ist ein Platzhalter für Codezeilen aus Listing 5, die genau hier eingebaut werden können.

```

If intCnt = 0 Then
    MsgBox "Keine Akronyme im relevanten Suchbereich gefunden", vbExclamation
    GoTo ExitPoint ' Programm abbrechen
End If
' Vorbereitete Tabellenzeilen hinter "Abkürzungsverzeichnis" einfügen
docTgt.Bookmarks(conBkmNm).Select
With Selection
    .TypeParagraph
    .InsertBefore strOutput
End With
' Vorbereitete Tabellenzeilen in eine neue Tabelle mit 3 Spalten umwandeln
Set newTbl = Selection.ConvertToTable( _
    Separator:=wdSeparateByTabs, _
    NumRows:=Selection.Paragraphs.Count, _
    NumColumns:=3)
' Neue Tabelle der Akronyme ggf. sortieren
With newTbl
    If .Rows.Count > 1 Then
        .Sort ExcludeHeader:=True, FieldNumber:="Column 1", _
            SortFieldType:=wdSortFieldAlphanumeric, _
            SortOrder:=wdSortOrderAscending
    End If
End With
' Neue Tabelle der Akronyme formatieren
Call FormatNewTable(newTbl)
' Bereich der neuen Tabelle der Akronyme auswählen
Set rngTbl = newTbl.Range
rngTbl.Select
With Selection
    ' Textmarke der eingefügten Tabelle erneuern
    .Bookmarks.Add Range:=rngTbl, Name:=conBkmNm
    Do ' Leere Absätze unter der eingefügten Tabelle löschen, falls vorhanden
        bolEmpty = False
        .MoveDown unit:=wdLine, Count:=1
        If .Paragraphs(1).Range.Text = vbCr Then
            .Range.Delete
            bolEmpty = True
        End If
    Loop Until Not bolEmpty
    .GoTo What:=wdGoToBookmark, Name:=conBkmNm
    Do ' Leere Absätze über der eingefügten Tabelle löschen, falls vorhanden.
        bolEmpty = False
        .MoveUp unit:=wdLine, Count:=1
        If .Paragraphs(1).Range.Text = vbCr Then
            .Range.Delete
            bolEmpty = True
        End If
    Loop Until Not bolEmpty
End With
ExitPoint:
On Error Resume Next
Application.ScreenUpdating = True ' Bildschirmaktualisierung einschalten
Selection.HomeKey unit:=wdStory
MsgBox "Fertig: " & Str(intCnt) & " Akronyme gefunden." & vbCrLf & _
    "Tabelle der Akronyme in das Abkürzungsverzeichnis des Zieldokuments " & vbCrLf & _
    docTgt.FullName & vbCrLf & _
    "eingefügt.", vbInformation
Set rngTbl = Nothing
Set rngPara = Nothing
Set rngSctn = Nothing
Set rngResult = Nothing
Exit Sub
ErrorHandler:
MsgBox "Fehler " & Err.Description, vbCritical
Resume ' ExitPoint
End Sub

```

Listing 4: Gefundene Akronyme tabellieren, Teil 3 von 3

Die in der Zeichenkette *strOutput* gesammelten Informationen werden an definierter Stelle hinter der Überschrift *Abkürzungsverzeichnis* im Zieldokument in eine Word-Tabelle umge-

wandelt. Diese Tabelle wird mit dem Unterprogramm *FormatNewTable* aufbereitet. Anschließend wird im Unterprogramm aufgeräumt

```
strExplain = vbNullString ' Anfangswert für die Bedeutung eines Akronyms
' Schreibweise im Text: ACRONYM (Bedeutung)
With Selection
    .MoveRight unit:=wdWord
    .MoveRight unit:=wdCharacter, Extend:=wdExtend
    If .Text = "(" Then
        Do While .Text <> ")"
            strExplain = strExplain & .Text
            .Collapse Direction:=wdCollapseEnd
            .MoveRight unit:=wdCharacter, Extend:=wdExtend
        Loop
    End If
    .Collapse Direction:=wdCollapseEnd
    If Left(strExplain, 1) = "(" Then
        strExplain = Mid(strExplain, 2, Len(strExplain))
    End If
    ' Schreibweise im Text: (Bedeutung) ACRONYM
    If Len(strExplain) = 0 Then ' Wenn Bedeutung noch nicht gefunden, dann ...
        .MoveLeft unit:=wdCharacter, Count:=1, Extend:=wdMove
        .MoveLeft unit:=wdWord
        .MoveLeft unit:=wdCharacter, Count:=2, Extend:=wdMove
        If .Text = ")" Then
            Do While .Text <> "("
                strExplain = strExplain & .Text
                .Collapse Direction:=wdCollapseStart
                .MoveLeft unit:=wdCharacter, Extend:=wdExtend
            Loop
        End If
        .Collapse Direction:=wdCollapseEnd
        If Left(strExplain, 1) = ")" Then
            strExplain = StrReverse(Mid(strExplain, 2, Len(strExplain)))
        End If
    End If
End With
End With
```

Listing 5: Bedeutung der Akronyme bei bestimmter Schreibweise im Text ermitteln⁷

Die in Listing 5 enthaltenen Codezeilen sind optional. Sie können an der grau markierten Stelle in Listing 3 eingefügt werden.

⁷ Die Codezeilen für die Schreibweise Akronym (Bedeutung) wurden erstellt in Anlehnung an (Wyatt, 2015).

```

Sub DeleteOldTable(docTgt As Document, strBkm As String)
' Alte Tabelle im Zieldokument (docTgt) löschen,
' falls die zugehörige Textmarke (strBkm) existiert,
' sonst die Textmarke erneuern.
With docTgt.Bookmarks
  If .Exists(strBkm) Then ' Wenn Textmarke existiert, ...
    Selection.GoTo What:=wdGoToBookmark, Name:=strBkm
    If Selection.Information(wdWithInTable) Then
      Selection.Tables(1).Delete ' alte Tabelle löschen
    End If
  Else ' sonst Textmarke für Tabelle der Akronyme setzen
    Selection.TypeParagraph
    .Add Name:=strBkm, Range:=Selection.Range
    .DefaultSorting = wdSortByName
    .ShowHidden = False
  End If
End With
End Sub

```

Listing 6: Alte Tabelle der Akronyme löschen und zugehörige Textmarke erneuern

Das Programm *DeleteOldTable* ist wichtig für die Wiederholbarkeit der Suche nach Akronymen im Zieldokument.

```

Sub FormatNewTable(tblNew As Table)
Dim objCell As Cell
Dim lngRow As Long
With tblNew
  .Rows.Alignment = wdAlignRowCenter ' Tabellenzeilen zentrieren
  If .Style <> "Tabellenraster" Then
    .Style = "Tabellenraster"
  End If
  With .Rows(1)
    .HeadingFormat = True ' als Kopfzeile formatieren
    With .Range
      .Shading.BackgroundPatternColor = wdColorGray25 ' Kopfzeile hellgrau schattieren
      .Font.Bold = True ' Kopfzeile fett drucken
    End With
  End With
  For lngRow = 2 To .Rows.Count
    For Each objCell In .Rows(lngRow).Cells
      ' Schriftgrad auf 10 Punkte festlegen
      objCell.Range.Font.Size = 10
    Next objCell
  Next lngRow
  ' Seitennummern in der letzten Tabellenspalte horizontal und vertikal zentrieren
  For Each objCell In .Columns(.Columns.Count).Cells
    With objCell
      .Range.ParagraphFormat.Alignment = wdAlignParagraphCenter
      .VerticalAlignment = wdCellAlignVerticalCenter
    End With
  Next objCell
  ' Breite der drei Tabellenspalten prozentual festlegen
  .Columns.PreferredWidthType = wdPreferredWidthPercent
  .Columns(1).PreferredWidth = 25
  .Columns(2).PreferredWidth = 60
  .Columns(3).PreferredWidth = 15
End With
End Sub

```

Listing 7: Neue Tabelle der Akronyme formatieren

Das Unterprogramm *FormatNewTable* führt die in Listing 7 gelb markierten Aufgaben durch.

```
Function OpenTargetFile(strFileNm As String)
    With Application.Dialogs(wdDialogFileOpen) ' Eingebauter Word-Dialog: Datei öffnen
        .Name = strFileNm
        Options.DefaultOpenFormat = wdOpenFormatAllWord
        If .Display Then
            OpenTargetFile = .Name
        Else
            OpenTargetFile = ""
        End If
    End With
End Function
```

Listing 8: Auswahl der Zieldatei mit eingebautem Word-Dialog

```

Sub FindListOfAcronyms ()
Const conExt      As String = ".docx" ' Dateizusatz
Dim docSrc        As Document      ' Quelldokument
Dim docTgt        As Document      ' Zieldokument
Dim lngSctn       As Long           ' Index-Nr. d. relevanten Abschnitts im Zieldokument
Dim strTgtNm      As String         ' Name des Zieldokuments
Dim strTgtPath    As String         ' Pfad zum Zieldokument
Dim strTgtFullNm  As String         ' voller Name des Zieldokuments
Dim strErrMsg     As String         ' Fehlermeldung
On Error GoTo ErrHandler           ' Fehlerbehandlungsroutine
Set docSrc = ActiveDocument        ' Verweis zum Quelldokument setzen
strTgtPath = docSrc.Path           ' Pfad zum Zieldokument bestimmen
If Right(strTgtPath, 1) <> "\" Then strTgtPath = strTgtPath + "\"
strTgtNm = OpenTargetFile(strTgtPath &
    "*" & conExt)                 ' Zieldatei mit Word-Dialog öffnen
If Len(strTgtNm) > 0 Then
    strTgtFullNm = strTgtPath & strTgtNm
Else
    MsgBox "Es wurde keine Zieldatei ausgewählt.", vbInformation
    GoTo ExitHandler
End If
Documents.Open FileName:=strTgtFullNm ' Zieldatei öffnen
Set docTgt = ActiveDocument           ' Verweis zum Zieldokument setzen
docTgt.Select                         ' Zieldokument auswählen
If docTgt.Sections.Count < 2 Then
    MsgBox "Das Zieldokument enthält nur 1 Abschnitt (Section).", vbInformation
    GoTo ExitHandler
End If
Application.ScreenUpdating = False    ' Bildschirmaktualisierung ausschalten
With Selection.Find                   ' 'Abkürzungsverzeichnis' im Zieldokument finden
    .ClearFormatting
    .Forward = True
    .Format = True
    .Style = "ÜberschriftOhneNr"     ' benutzerdefinierte Formatvorlage
    .MatchCase = True
    .MatchWholeWord = True
    .Wrap = wdFindStop
    .Execute FindText:="Abkürzungsverzeichnis"
    If .Found = True Then             ' Wenn Suche erfolgreich, dann ...
        Selection.Collapse wdCollapseEnd ' Ergebnis d. Suche auf Endpunkt reduzieren
        Call ListAcronyms(docTgt)     ' Tabelle Akronyme im Zieldokument erstellen
    End If
End With
ExitHandler:
On Error Resume Next
If docTgt.Saved = False Then docTgt.Save ' Zieldokument speichern, aber nicht schließen
Set docTgt = Nothing                   ' Verbindung zum Zieldokument aufheben
Application.VBE.MainWindow.Visible = False ' VBA-Editor ausblenden
docSrc.Close SaveChanges:=wdSaveChanges ' Quelldokument schliessen und speichern
Set docSrc = Nothing                 ' Verbindung zum Quelldokument aufheben
Application.ScreenUpdating = True     ' Bildschirmaktualisierung einschalten
Exit Sub                               ' Programm verlassen
ErrHandler:
' Fehlermeldung zusammensetzen
strErrMsg = "Laufzeitfehler: # " & Str(Err.Number) & _
    " wurde ausgelöst von " & _
    & Err.Source & vbCr & Err.Description
' Fehlermeldung anzeigen
MsgBox strErrMsg, vbCritical, "Laufzeitfehler", Err.HelpFile, Err.HelpContext
Resume ExitHandler ' Programm bei Zeilenmarke "ExitHandler" fortsetzen
End Sub

```

Listing 9: Ersatz f. Listing 1 bei Auswahl d. Zieldatei mit eingebautem Word-Dialog

Literatur-/Quellenverzeichnis

Fredborg, L. (2008). *Extract ACRONYMS to New Document*. Abgerufen am 10. 7 2016 von http://www.thedoctools.com/downloads/basACRONYMS_Extract.htm

Mayor, G. (2012). *Idiot's guide to installing macros*. Abgerufen am 29. 6 2016 von Graham Mayor - Word Pages: http://www.gmayor.com/installing_macro.htm

o.V. (21. 6 2016). *Akronym*. Abgerufen am 2. 7 2016 von WIKIPEDIA:
<https://de.wikipedia.org/wiki/Akronym>

Puls, Erika. (13. 6 2013). *Wissenschaftliche Arbeit #1 mit Word 2010 formatieren*. (HAWK)
Abgerufen am 29. 6 2016 von <https://www.youtube.com/watch?v=e53GMQpZAzE>

Wyatt, A. (22. 6 2015). *Auto Creation of an Acronym List*. Abgerufen am 2. 7 2016 von Allen Wyatt's WordTips:
http://wordribbon.tips.net/T010935_Auto_Creation_of_an_Acronym_List.html