

Einige Probleme mit Word und deren Lösung

Inhalt

	Seite
Vorwort	3
1 Problem mit Silbentrennzeichen und "Geschützter Bindestrich" beim Einfügen von "Nur Text"	3
2 Einige Problem mit der Rechtschreibprüfung in Word 2010	4
2.1 Fehlermeldung bei Silbentrennungen in Word 2010	4
2.2 Lösung für "Feststehende Begriffe" in Word 2010	4
2.3 Keine Lösung für Grammatikfehlermeldung bei dem Geschlecht von Abkürzungen	4
3 Fehler durch die AutoKorrektur	5
3.1 Einfacher Fall	5
3.2 Fehler in der AutoKorrektur durch falsche Spracherkennung	6
3.3 Fehler auch ohne falschen Eintrag bei der AutoKorrektur	7
3.4 Linienerzeugung durch "AutoFormat während der Eingabe"	8
4 Probleme und Lösungen beim Inhaltsverzeichnis	9
4.1 Bedeutung der Schalter im Feldcode des Inhaltsverzeichnisses	9
4.2 Fehlende Anklickbarkeit nach Konvertierung in Pdf mittels Adobe Acrobat X (Inhaltsverzeichnis auch in Pdf-Datei anklickbar machen)	9
4.3 Verlorene Zeichen-Eigenschaften bei gesetztem "Schalter h",	12
4.4 Problem mit Absätzen des Formats "Standard" im Inhaltsverzeichnis	13
5 Probleme in Word 2000/Word 2003 mit den Fonts "Arial Unicode" und "Cambria Math"	15
5.1 Buchstabenabstände bei Arial Unicode	15
5.2 Zeilenabstände bei Cambria Math	15
6 NumLock-Taste deaktiviert sich in Word 2010 nach Anwendung von SendKeys -Befehlen	16
6.1 aufwendige, aber zuverlässige Lösung, um eine Menüband-Registerkarte (Ribbon-Tab) in Word 2010 per VBA-Makro anzusprechen	16
6.2 CapsLock wieder-ausschalten und NumLock wieder-einschalten	19
7 Probleme mit Windows 7	20
7.1 Problem mit sich drehendem Bildschirm in Windows 7	20
7.2 Das Makro für den Briefkopf mit aktuellem Datum "Sub AutoNew()" funktioniert unter Windows 7 nicht mehr zuverlässig	20
7.3 Verlust des Inhalts der Zwischenablage beim Erzeugen einer neuen Datei	22
8 Problem mit dem Entlinken von Feldern in Tabellen	23
9 Einstellungen von Word sichern beim Übertragen der persönlichen Word-Einstellungen auf einen neuen Computer	24
9.1 Wo ist die Normal.dotm versteckt, und wo ist der StartUp-Ordner?	24
9.2 Bilder-Einfüge-Pfad kann nicht mehr voreingestellt werden	24
9.3 Per Hand zu übertragende Einstellungen	25
9.4 Formblatt mit Makros zur Übertragung der Tastenkürzel und der (deutschen) AutoKorrekturen auf einen neuen Computer	27
10 Probleme mit falsch verlinkten Bildern	29
10.1 Fehlende Bilder	29
10.2 Bei Word 2010 herkömmliche Feldfunktion erzwingen	30
10.3 Bilder der falschen Datei bei Word 2010	30
11 Formeln mit dem Formel-Editor OMML in Word 2010	31
11.1 Aufruf des Formeleditor per Button oder per Makrobefehl	31

	Seite
11.2 Die Anwendung des Formeleditors in Word 2010	33
11.3 Anwendungsbeispiel für den Formeleditor in Word 2010: Grenzabmaße übereinander anordnen	39
11.4 Beispiel, in dem der Formeleditor einige Zeichen in Sonderzeichen einer höheren Unicode-Ebene umwandelte.....	40
12 Word-Dateien mittels Word 2010 in eine Pdf-Datei mit "Lesezeichen" umwandeln.....	41
12.1 Problem beim Suchen nach Begriffen in Pdf-Dateien, die mit Adobe-Acrobat erstellt wurden.....	41
12.2 Lesezeichen erzeugen beim Erstellen einer Pdf-Datei mittels Word-Funktion"	41
12.3 Inhaltsverzeichnis bei den Lesezeichen der Pdf-Datei ergänzen	42
12.4 Absätze auf eine andere Art für die "Lesezeichen" präparieren	43
12.5 Screenshot der Lesezeichen in der Pdf-Datei	44
13 Bildschirmansicht invertieren mit "Weiß auf Schwarz"	45
13.1 Weiß auf Schwarz in Word 2010	45
13.2 Weiß auf Blau in Word für ältere Apple-Computer.....	45
13.3 Weiß auf Blau in Word 2000 bis Word 2003	46
14 Zwei Word-Dokumente auf dem Bildschirm vergleichen	47
15 Bildschirmtastatur	47
15.1 Aufruf der Windows-Bildschirmtastatur unter Windows XP	47
15.2 Bildschirmtastatur unter Windows 7.....	47
15.3 Bildschirmtastatur unter Windows 10.....	47
15.4 Abhilfe, wenn in der Bildschirmtastatur die Zehnertastatur fehlt.....	48
15.5 Makro für den Aufruf der Bildschirmtastatur	49
Anhang A Problem mit der (mir lästigen) Änderungsverfolgung und fehlende Anzeige in der Statusleiste von Word 2010.....	50
Anhang B Einige (unnötige) Makros für die AutoKorrektur	52
B.1 Makros zum Entfernen von AutoKorrekturen	52
B.2 ".acl"-Dateien suchen und erklären.....	55
B.3 Die Makros im Formblatt zur Übertragung von Tastenkürzeln und AutoKorrekturen nach Abschnitt 9.4	57
Anhang C Anleitung: Eigene Buttons in der Menüleiste (Ribbon) oder in der Schnellstartleiste (QAT) von Word 2010 einfügen.....	77
C.1 Allgemeine Anleitung.....	77
C.2 Buttons in der Schnellstartleiste einer Dokumentvorlage (".dotm"-Datei), die nicht die "Normal.dotm" ist	78
C.3 Buttons in der Menüleiste einer ".dotm"-Datei (Quelltext-Bearbeitung)	78
C.4 Vorhandene Ribbon-Buttons in einer Dokumentvorlage ausblenden und ersetzen (Quelltext-Bearbeitung).....	84
C.5 Tastenkürzel für Buttons und auch für Ribbon-Tabs zuordnen.....	86
C.6 Buttons in einer vorhandenen Registerkarte (Ribbon-Tab) ergänzen.....	89
C.7 Weitere Tipps, Namen der eingebauten Ribbons und einige Button-Farben	90
C.8 Alte Symbolleisten von Word 2000 usw. in Word 2010 entfernen	92
C.9 Vermeintliches Problem "Menüband ist verschwunden"	93
Weitere Literatur	94
Impressum.....	94

Vorwort

In diesem Aufsatz habe ich einige Probleme mit Word aufgelistet. Die meisten davon lassen sich lösen, siehe die folgenden Abschnitte.

Insgesamt ist Word im Laufe der Jahre besser geworden. Meine Liste der Probleme ist wesentlich kürzer geworden als früher. Auch "AutoTexte" werden in Word 2010 jetzt wieder besser unterstützt als bei dessen Einführung und es entstehen nicht mehr so leicht "Kästchen", wenn die Schriftart fehlt.

Allerdings fehlt mir die aus der alten Word-Version gewohnte eingebaute "Hilfe", für die ich bei Word 2010 jetzt ins Internet gehen muss. Und wenn ich in einer Tabelle einen Ausschnitt markieren will mit Alt + Maus, dann öffnet sich eine Internet-Suchfunktion "Recherchieren", die ich bis jetzt noch nicht deaktivieren konnte (*das Problem fand ich im Internet, aber dort gab es keine Lösungen dafür. Über eine Zuschrift mit einer Lösung hierfür wäre ich dankbar*).

H. Chris Gast, Stand Juli 2019

1 Problem mit Silbentrennzeichen und "Geschützter Bindestrich" beim Einfügen von "Nur Text"

Wenn man Text mit Silbentrennung kopiert und an anderer Stelle **als Inhalt** wieder einfügt in Word 2010 (*rechte Maustaste, Einfügeoptionen* → *Nur Text*), wird aus den unsichtbaren Silbentrennstrichen ("Bedingter Trennstrich") seit Frühjahr 2015 das mathematische **Nicht-Zeichen** "–", und am Zeilenende der einfache Bindestrich.

Dieses Nicht-Zeichen ist leider auch auf dem Papier-Ausdruck und in Pdf-Dateien noch zu sehen.

Ein ähnliches Problem gibt es beim Geschützten Bindestrich (Fester Bindestrich), dieser verschwindet ersatzlos beim Inhalt-Einfügen.

Der dezimale Code der betroffenen Zeichen:

Bedingter Trennstrich ("SOFT HYPHEN):	Code "031" (Steuerzeichen);
Nicht-Zeichen ("NOT SIGN"):	Code "0172";
Bindestrich ("HYPHEN-MINUS"):	Code "045";
Geschützter Bindestrich ("NON-BREAKING HYPHEN"):	Code "030" (Steuerzeichen).

Dafür biete ich als Lösung in [3] einige passende Word-Makros mit korrigiertem "Inhalt-Einfügen" an.

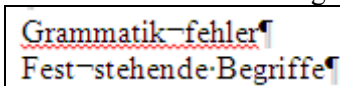
2 Einige Problem mit der Rechtschreibprüfung in Word 2010

In Word 2010 ist die Rechtschreibprüfung schwieriger geworden. Silbentrennung wird oft (aber nicht immer) als Fehler angezeigt. Am Ende der Zeile kann man nicht mehr sehen, ob ein bedingter Strich oder ein normaler Bindestrich steht (siehe Abschnitt 2.1). Feststehende Begriffe lassen sich nicht in der Rechtschreibprüfung speichern, siehe Abschnitt 2.2).

Die meisten Fehlermeldungen der Rechtschreibprüfung beruhen auf falscher Spracherkennung. Man kann die automatische Spracherkennung abschalten, aber andererseits ist diese auch recht nützlich. In der Regel muss man mindestens einen ganzen Satz oder Absatz schreiben, bevor Word eine Sprache erkennt.

2.1 Fehlermeldung bei Silbentrennungen in Word 2010

Die Silbentrennung wird in Word 2010 manchmal als Fehler angezeigt, aber nicht immer.



Man kann häufig verwendete Silbentrennungen allerdings ins Wörterbuch aufnehmen.

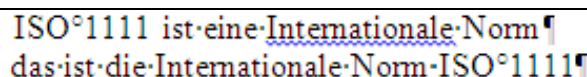
Ferner kann man am Zeilen-Ende bei Word 2010 nicht erkennen, ob eine Silbentrennung oder ein normaler Bindestrich verwendet wurde. Das stört aber nur beim Korrekturlesen, nicht beim normalen Texte-Schreiben.

2.2 Lösung für "Feststehende Begriffe" in Word 2010

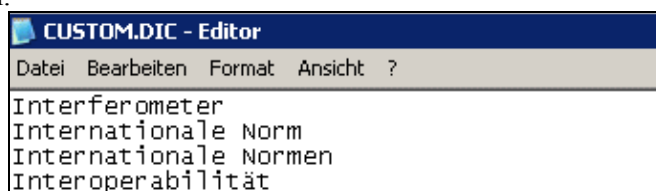
Problem: Word 2010 meldet bei Feststehenden Begriffen wie "Die Internationale Norm ISO 111" einen Fehler (blaue Wellenlinie). Dies ließ sich auch nicht durch "Hinzufügen zum Wörterbuch" korrigieren¹.

Im Internet wird die Rechtschreibprüfung vom Duden für 19,95 Euro angepriesen, die "Feststehende Begriffe" anbietet. Eine andere Quelle² empfiehlt die Software "PhraseExpress" (<http://www.phraseexpress.com/de>).

Meine Lösung: Ich fügte Feststehende Begriffe wie "Die Internationale Norm ISO 1111" von Hand in das Benutzerwörterbuch "CUSTOM.DIC" ein.



"Internationale Normen" werden beim mir manchmal noch angekringelt, manchmal nicht. Wahrscheinlich müsste ich jetzt noch die Kombination "eine Internationale Norm" zusätzlich in das Wörterbuch aufnehmen.



2.3 Keine Lösung für Grammatikfehlermeldung bei dem Geschlecht von Abkürzungen

Word 2010 meldet bei dem Artikel von einigen ungebräuchlichen Abkürzungen einen Grammatikfehler (Grüne Wellenlinie) und schlägt einen anderen Artikel vor. Hierfür fand ich bis jetzt (Juni 2012) keine Lösung.

¹ Angeblich soll es dadurch möglich sein, dass man Adjektiv und Substantiv markiert und dann erst Rechtsklick mit der Maus macht [mündliche Auskunft eines Bekannten]. Bei mir funktioniert das aber so nicht.

² Frau Richter, <http://www.schreibbuero-richter.de>

3 Fehler durch die **AutoKorrektur**

3.1 Einfacher Fall

Die AutoKorrektur beseitigt nicht nur Fehler, sondern manchmal erzeugt sie auch Fehler. Man schreibt zum Beispiel "DNA", und die AutoKorrektur macht daraus "DANN"³.

Abhilfe:

Man schreibe DNA, drücke dann kein Leerzeichen oder Absatzmarke, sondern klicke in Word 2000 und Word 2003 sofort den Pfad an: "E_xtras → AutoKorrektur...", siehe Bild.

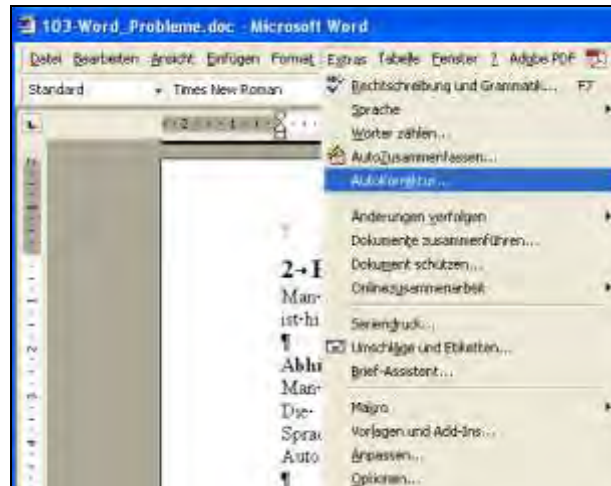


Bild 3.1: Pfad zur AutoKorrektur in Word 2000 bis Word 2003

Und in **Word 2010**: "Datei → Optionen → Dokumentprüfung → AutoKorrektur-Optionen"
Dort sieht man dann die vorgesehene "Verbesserung" (siehe folgendes Bild).

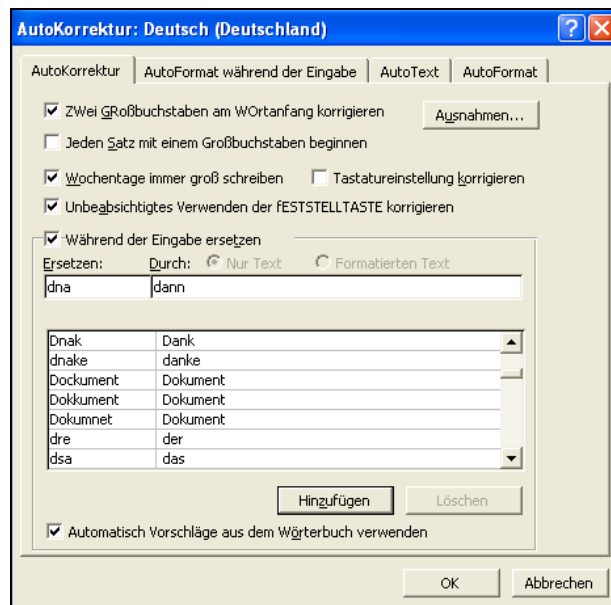


Bild 3.2: Zu löschende Einträge in der AutoKorrektur

Man markiere die entsprechende Zeile und lösche sie (Button rechts darunter, im obigen Screenshot noch grau). Vielleicht sollte man bei der Gelegenheit die ganze Liste mal in Ruhe durchsehen.

³ Hinweis einer Freundin, die Word auf einem Apple-Computer benutzt. Der Fehler tritt aber auch in Word 2000 und Word 2010 unter Windows auf.

In **Word 2010** findet man die AutoKorrektur unter "Datei → Optionen → Dokumentprüfung → AutoKorrektur-Optionen" oder man setze den Cursor auf das unterkringelte Wort und klicke mit der rechten Maustaste auf "AutoKorrektur".

3.2 Fehler in der AutoKorrektur durch falsche Spracherkennung

Bei der Rechtschreibprüfung kann man mit falscher Spracherkennung noch gut leben. Das Schlimme an der falschen Spracherkennung ist, dass auch die AutoKorrektur davon betroffen ist. Und die verfälscht den eingegebenen Text.

Richtig geschriebene Texte werden manchmal beim Schreiben durch die AutoKorrektur mit falscher Spracherkennung verfälscht, zum Beispiel:

- a) im Englischen wird zum Beispiel 'of the' durch 'oft he' ersetzt, obwohl es das Wort "oft" im Englischen nicht gibt.
- b) in deutschen Sätzen wird das Verb "werden" durch "warden" ersetzt, obwohl es dieses Wort im Deutschen nicht gibt.

Einfache Lösung

Eine der drei folgenden Maßnahmen wählen:

- 1) Automatische Spracherkennung ausschalten:
 - In Word 2000 und Word 2003 unter "Extras → Sprache → Sprache bestimmen" und dort **Haken rausnehmen** bei "Sprache automatisch erkennen";
 - In Word 2010 unten in der Statusleiste auf die angegebene Sprache klicken und im dann erscheinenden Fenster den **Haken rausnehmen** bei "Sprache automatisch erkennen," oder den Reiter (Ribbon) "Überprüfen" anklicken, dann im Bereich "Sprache" auf "Sprache für die Korrekturhilfen festlegen..." klicken, dort Haken rausnehmen bei "Sprache automatisch erkennen" und vielleicht dort auch die "Spracheinstellungen" etwas entrümpeln.
- 2) Entfernung aller AutoKorrektur-Einträge der nichtgebrauchten Sprachen (notfalls per Makro).
- 3) AutoKorrektur gänzlich abschalten (Pfad in Word 2000 und Word 2003: "Extras → AutoKorrektur" bzw. in Word 2010 unter: "Datei → Optionen → Dokumentprüfung → AutoKorrektur-Optionen" Dort dann den **Haken rausnehmen** bei "Während der Eingabe ersetzen".

Da ich einige Ersetzungen der Autokorrektur sehr nützlich finde, möchte ich aber nur die störenden Eigenschaften beseitigen.

Maßgeschneiderte Lösung:

Ich entfernte in der **deutschen** AutoKorrektur die Ersetzung von "of" durch "oft".

Dazu Sprache auf Deutsch stellen bzw. den Cursor auf ein als "Deutsch" erkanntes Stück Text stellen;

einen Fehler wie "aaa" schreiben, der von Word rot unterkringelt wird;

dann Rechtsklick mit der Maus, AutoKorrektur → "AutoKorrektur-Optionen...";⁴

Dort löscht man in der deutschen AutoKorrektur die entsprechenden Einträge.

⁴ Andere Möglichkeit:

– in Word 2000 und Word 2003: "Extras → AutoKorrektur...".

– in Word 2010: "Datei → Optionen → Dokumentprüfung → AutoKorrektur-Optionen".

Aber auch hier muss der Cursor in einem Textstück mit der betreffenden "Sprache" stehen.

Entsprechend beseitigte ich im Englischen den Eintrag, der aus "wurden" oder "werden" das falsche Wort "warden" erzeugte:

Sprache auf Englisch stellen;

einen Fehler schreiben wie "aaa";

Rechtsklick mit der Maus, AutoKorrektur → "AutoKorrektur-Optionen..."

Damit kommt man in die englischen AutoKorrektur-Optionen. Dort löscht man in der englischen AutoKorrektur die Einträge, die "warden" erzeugten.

3.3 Fehler auch ohne falschen Eintrag bei der AutoKorrektur

Das obige Verfahren half aber nicht bei allen Falsch-Ersetzungen! Zum Teil erfolgten auch AutoKorrekturen bei Wörtern, die gar nicht in der AutoKorrektur-Liste vorhanden sind!

Dafür fand ich folgende Lösung:

Man muss in den AutoKorrektur-Optionen in Word 2010 im Pfad:

"Datei → Optionen → Dokumentprüfung → AutoKorrektur-Optionen"

unten links den Haken herausnehmen (siehe folgendes Bild).

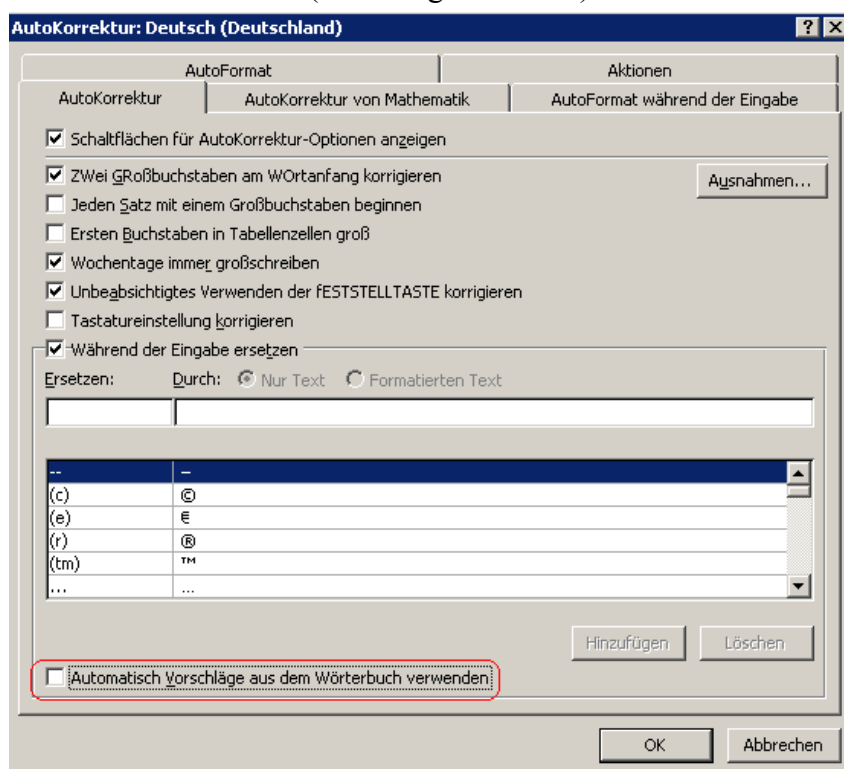


Bild 3.3: Zu löschender Haken in den AutoKorrektur-Optionen

Grund: "Beim Installieren von Microsoft Office wird das Feature AutoKorrektur standardmäßig so eingerichtet, dass der Versuch unternommen wird, ein falsch geschriebenes Wort mit einem Wort im von der Rechtschreibprüfung verwendeten **Hauptwörterbuch** abzugleichen und zu korrigieren..." [Quelle: Hilfe von Word 2010, April 2012]

Erst, nachdem ich [unten links](#) (siehe Bild oben) den Haken rausnahm, gab es keine Verfälschungen durch die AutoKorrektur mehr. Stattdessen kann man auch das Wörterbuch ("Custom.dic") überprüfen und dort falsche Einträge entfernen. Dies hilft aber **nicht** bei **mehrsprachigen** Texten und **falscher Spracherkennung**.

3.4 Linienerzeugung durch "AutoFormat während der Eingabe"

Word kann auf einfache Art Linien erzeugen, wenn dies in der AutoKorrektur so eingeschaltet ist:

Word 2000 und Word 2003:

"Extras" → "AutoKorrektur-Optionen..." → "Autoformat während der Eingabe" → "Rahmenlinien"

in Word 2010:

"Datei" → "Optionen" → "Dokumentprüfung" → "AutoKorrektur-Optionen" →

"Autoformat während der Eingabe" → "Rahmenlinien"

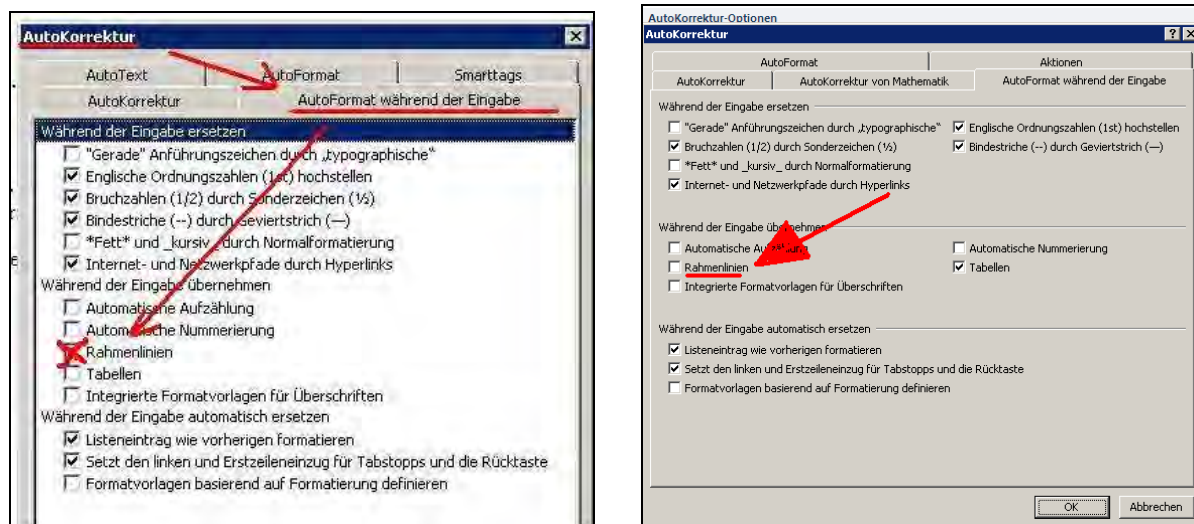


Bild 3.4: Links Word 2000 bzw. Word 2003, rechts Word 2010

Beispiele:

- dann Enter ergibt eine einfache Linie;
- === dann Enter ergibt eine Doppellinie;
- *** dann Enter ergibt eine gepunktete Linie.

Ich selber benutze diese Funktionen **nie**. Diese Linien lassen sich nur mit Mühe löschen!

Löschen von solchen Linien: Man muss dazu die Linienzeile und die Absätze mit Text darüber und darunter markieren und dann mit dem Befehl Strg + Q von allen Formatierungen befreien. Markiert man dazu nur die Leerzeilen davor und danach, springt unter Umständen die Linie einfach nur weiter nach oben.

4 Probleme und Lösungen beim Inhaltsverzeichnis

4.1 Bedeutung der **Schalter** im Feldcode des Inhaltsverzeichnisses

`\b` = Word legt nur einen Index für den Bereich des Dokumentes an, dem Sie Textmarke zugewiesen haben.

`\c` = "SEQBezeichner" Listet Bilder, Tabellen usw. auf, die mit Hilfe eines SEQ-Feldes nummeriert wurden.

`\f` = für die Auswahl von "TC-Feldern", die mit Alt + Shift + O erzeugt werden,
z. B. `{TOC \o "1-3" \f. . . }`

`\o` = Verzeichnis, das sich an den Überschriftformatvorlagen orientiert und die Überschriftenebenen berücksichtigt, z. B. `{TOC \o "1-3" . . . }`

`\h` = **Hyperlink** **wichtig für Adobe X**, wenn das Inhaltsverzeichnis auch in der PDF-Datei anklickbar sein soll (2. Februar 2012).

PROBLEM: Bei Schalter "h" gehen die Schriftattribute "**Fett**" bzw. "Nicht-fett", "Unterstrichungen" und "**Schriftfarben**" verloren! Nur "**Markierungen**" und "**kursiv**" bleiben erhalten.

`\t` = Für selbst definierte Formatvorlagen

Beispiel: `{ TOC \o "1-2" \h \z \t "Überschrift_3ohne;3" }` für eigene Aufsätze
oder `{ TOC \h \z \t "Figure title" \c }` für Bilder in Normen

`\u` = Nutzung der vorhandenen Gliederungsebenen für Absätze

`\w` = Behält Tabstoppszeichen in Verzeichniseinträgen

`\x` = Behält Zeilenendmarken innerhalb von Verzeichniseinträgen bei

`\z` = Blendet Füllzeichen und Seitenzahlen in der Weblayoutansicht aus.

4.2 Fehlende Anklickbarkeit

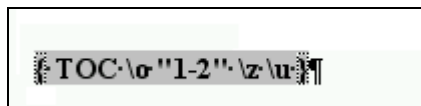
nach Konvertierung in Pdf mittels **Adobe Acrobat X**
(Inhaltsverzeichnis auch in Pdf-Datei anklickbar machen)

Problem:

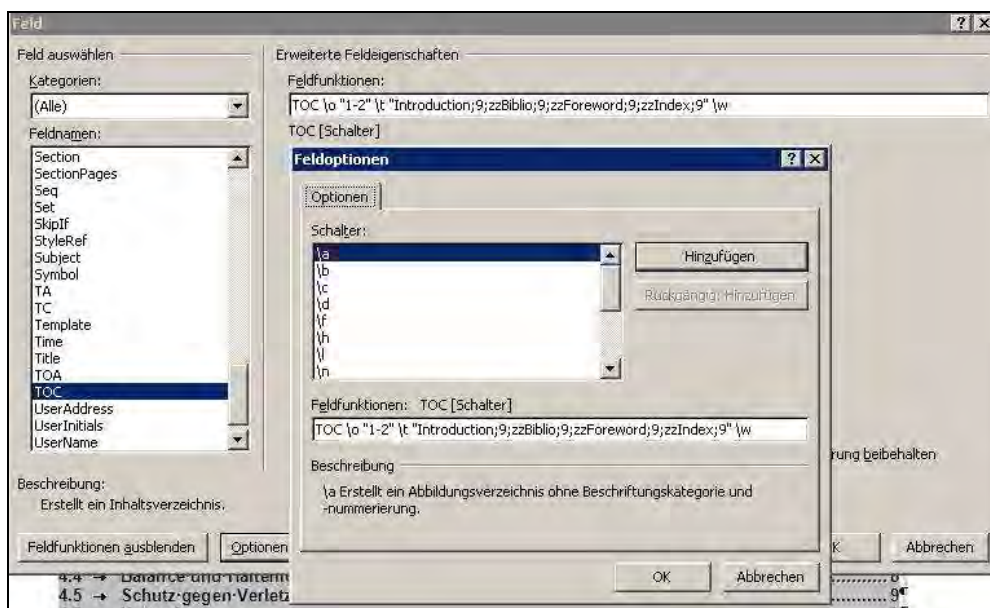
Bei Word 2010 in Verbindung mit Adobe Acrobat X werden manchmal die Inhaltsverzeichnisse in der Pdf-Datei nicht mehr anklickbar, während sie mit den früheren Adobe-Versionen anklickbar waren.

Grund:

Adobe Acrobat X verlangt die Anwesenheit eines Schalters "`\h`" im Feldcode, der für "Hyperlink" steht. Dieser fehlt hier im folgenden Screenshot:

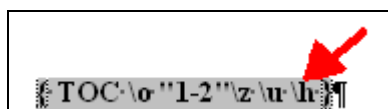


Dann im sich öffnenden Menü "Feldfunktionen" anklicken und anschließend "Optionen" anklicken.



Nun Schalter "\h" auswählen und "Hinzufügen".

Das Inhaltsverzeichnis ist nun für Adobe X tauglich und enthält den Schalter "\h" für "Hyperlink", siehe folgenden Screenshot eines Inhaltsverzeichnisses in der Feldcode-Ansicht (mit **Alt** + **F9**):



4.2.2 Alternativer Weg: Schalter "\h" in der Feldcode-Ansicht von Hand ergänzen (Word 2000 bis Word 2010)

Man kann in der Feldcode-Ansicht (**Alt** + **F9**) den Schalter "h" von Hand ergänzen, siehe Bild oben. Man muss dann aber anschließend noch das Inhaltsverzeichnis aktualisieren (Inhaltsverzeichnis markieren, **F9** drücken und "Gesamtes Verzeichnis aktualisieren" anklicken).

Wenn man unbedingt will, kann man auch ein Makro dazu verwenden, um den Schalter "h" (in der Feldcode-Ansicht) zu ergänzen und die Verzeichnisse zu aktualisieren:

Sub im_Inhaltsverzeichnis_direkt_Schalter_h_ergänzen()

```
' H. Chris Gast, 22. Mai 2012, Stand 17. Juni 2012.
' Dieses Makro ergänzt den Schalter "h" bei Inhalts- und Bildverzeichnissen, wenn er fehlt.
' Dabei ist es egal, wieviele Ebenen die Verzeichnisse
' oder auf welche Formatvorlagen die Verzeichnisse basieren.
' Ohne diesen Schalter "h" ist nämlich bei Verwendung von Adobe Acrobat X später in der
' Pdf-Datei das Inhaltsverzeichnis nicht mehr anklickbar. Dafür gehen bei Verwendung von
' Schalter "h" im Inhaltsverzeichnis die Eigenschaften "Schriftfarbe", "nicht-fett"
' und "unterstrichen" verloren.
```

Dim i As Integer

```
Zeile_Inhaltsverzeichnisse_aktualisieren:
```

```

For i = 1 To ActiveDocument.TablesOfContents.Count
ActiveDocument.TablesOfContents(i).Update
If InStr(ActiveDocument.TablesOfContents(i).Range.Fields(1).Code, "\h") = 0 Then
'
' Das folgende ergänzt in der Feldcode-Ansicht den "Schalter h".
' Wichtig ist hierbei das anschließende Updaten!
ActiveDocument.TablesOfContents(i).Range.Fields(1).Code.Text = _
ActiveDocument.TablesOfContents(i).Range.Fields(1).Code.Text & " \h "
ActiveDocument.TablesOfContents(i).Range.Fields(1).Code.Text = _
Replace(ActiveDocument.TablesOfContents(i).Range.Fields(1).Code.Text, " ", " ")
MsgBox "es fehlte im Verzeichnis der Schalter ""h"", der jetzt ergänzt wurde."
End If
ActiveDocument.TablesOfContents(i).Update
Next i
'
' Inhaltsverzeichnisse für Bilder und Tabellen:
For i = 1 To ActiveDocument.TablesOfFigures.Count
ActiveDocument.TablesOfFigures(i).Update
If InStr(ActiveDocument.TablesOfFigures(i).Range.Fields(1).Code, "\h") = 0 Then
ActiveDocument.TablesOfFigures(i).Range.Fields(1).Code.Text = _
ActiveDocument.TablesOfFigures(i).Range.Fields(1).Code.Text & " \h "
ActiveDocument.TablesOfFigures(i).Range.Fields(1).Code.Text = _
Replace(ActiveDocument.TablesOfFigures(i).Range.Fields(1).Code.Text, " ", " ")
End If
ActiveDocument.TablesOfFigures(i).Update
Next i
End Sub
'-----

```

4.3 Verlorene Zeichen-Eigenschaften bei gesetztem "Schalter h",

Die Schriftzeichen-Eigenschaften Schriftfarbe, "nicht-fett" (mager) und "unterstrichen" richten sich bei Anwesenheit des Schalter h nach der Formatvorlage für die Verzeichnisse, nicht nach den tatsächlichen Überschriftentexten.

Die Eigenschaft "kursiv" bleibt dagegen erhalten, auch die Farb-Markierungen (Texthervorhebungsfarbe).

In Word 2010 richtet sich die Schriftgröße im Inhaltsverzeichnis grundsätzlich nach der Formatvorlage für die Verzeichnisse. Dies empfinde ich aber als Vorteil.

4.4 Problem mit Absätzen des Formats "Standard" im Inhaltsverzeichnis

1.2→ Inhaltsverzeichnisse mit dem Fehler-Symptom¶		Seite¶
a) Richtig:¶		
1 → Korrektur, wenn "Standard"-Text im Inhaltsverzeichnis auftaucht		1¶
1.1→ Fehlerhafter Testtext aus einer E-Mail mit Outlook		1¶
1.2→ Inhaltsverzeichnisse mit dem Fehler-Symptom		1¶
1.3→ Diagnose		2¶
1.4→ Korrektur		2¶
Anhang A Testanhang		4¶
A.1→ Testüberschrift im Anhang		4¶
¶		
b) Falsch:¶		
1 → Korrektur, wenn "Standard"-Text im Inhaltsverzeichnis auftaucht		1¶
1.1→ Fehlerhafter Testtext aus einer E-Mail mit Outlook		1¶
Von: 'ccc@xxx.com'		1¶
Gesendet: 'Donnerstag, 30. April 2015 10:16'		1¶
An: 'ggg@yyy.de'		1¶
Betreff: 'Information zu xxx'		1¶
1.2→ Inhaltsverzeichnisse mit dem Fehler-Symptom		1¶
1.3→ Diagnose		2¶
1.4→ Korrektur		2¶
Anhang A Testanhang		4¶
A.1→ Testüberschrift im Anhang		4¶
¶		

Bild 4.1: Inhaltsverzeichnis mit fehlerhaften Einträgen (gelb markiert)

Beispiel: Fehlerhafter Testtext aus einer E-Mail mit Outlook

Die fehlerhaften Absätze habe ich hier gelb markiert, die Original-Adressen habe ich unkenntlich gemacht.

Von: ccc@xxx.com
 Gesendet: Donnerstag, 30. April 2015 10:16
 An: ggg@yyy.de
 Betreff: Information zu xxx

...
 Liebe ...,
 ...

4.4.1 Korrektur bei Absätzen mit "Standard" im Inhaltsverzeichnis

- a) Mit der Maus Rechtsklick auf den Absatz, der fälschlich im Inhaltsverzeichnis auftaucht, dann "Absatz" wählen, siehe Bild:

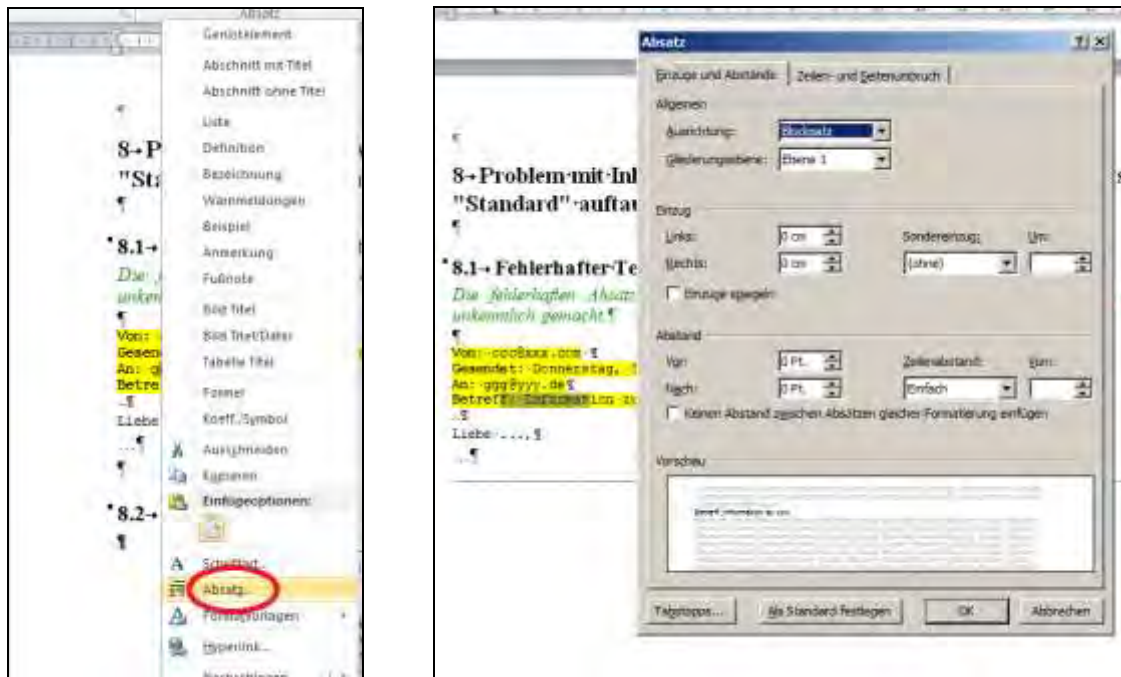


Bild 4.2: Rechtsklick, dann "Absatz..."

Als Gliederungsebene wird fälschlich die Gliederungsebene "Ebene 1" angezeigt. Diese ist zu ersetzen durch "Textkörper" (rechts neben "Ebene 1" den Pfeil nach unten anklicken und dann im Menü auswählen).

Sind mehrere Absätze betroffen, kann man auch die ganze Datei markieren und das obige ausführen. Echte Überschriften sind nicht veränderbar, so dass mit einem Schlag das gesamte Dokument bereinigt wird.

- b) Alternativ hilft bei mir auch, im Inhaltsverzeichnis den Schalter "u" herauszunehmen, siehe folgendes Bild:

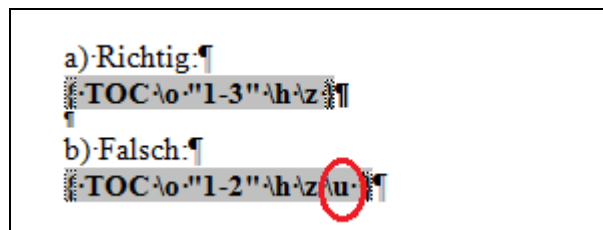


Bild 4.3: Schalter "u" aus Inhaltsverzeichnis herausnehmen

5 Probleme in **Word 2000/Word 2003** mit den Fonts "**Arial Unicode**" und "**Cambria Math**"

Nicht alles ist schlechter in Word 2010. Folgende Probleme gibt es in Word 2000, aber nicht mehr in Word 2010.

5.1 Buchstabenabstände bei Arial Unicode

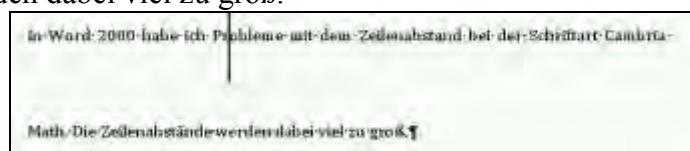
Wenn die Schriftart "Arial Unicode" bei nichtlateinischen Zeichen wie "αβγδε" auf einem Computer zu große Buchstabenabstände macht ("α β γ δ ε ¶"), sollte man keinesfalls an den Buchstabenabständen unter "Format → Zeichen herumbasteln, da dieser Fehler nicht auf allen Computern passiert. Sonst können nachher beim Empfänger der Datei oder beim Konvertieren in Pdf die Abstände zu gering werden: "αβδε". Der Fehler war (nach langer Suche) bei mir, dass ich noch unter "Extras → Optionen → Kompatibilität" die Einstellung "Word 97" hatte. Und Word 97 war noch nicht Unicode-tauglich.

Hier ist dann unter "Extras → Optionen → Kompatibilität" die Einstellung statt "Word 97" mindestens in "Word 2000" (oder in die Version, die man hat) zu ändern.

Oder man verzichtet hier auf Arial Unicode. Die in Word 2000 standardmäßig vorhandenen Schriftarten wie **Microsoft Sans Serif**, **Lucida Unicode**, **Tahoma** und **Palatino Linotype** reichen in den meisten Fällen (außer Chinesisch usw.) aus, wenn die verwendete Standard-Schriftart (bei mir Times New Roman) das gewünschte Zeichen nicht hat. Sogar den **Ligaturbogen** für zwei Buchstaben (U+0361 COMBINING DOUBLE INVERTED BREVE) wie in "āā". kann ich auch mit anderen Schriftart gut erzeugen, etwa mit DejaVu oder mit Roman Cyrillic. Und mit Cambria, wie es bei Word 2010 mitgeliefert wird, geht es natürlich auch in Word 2000 einwandfrei "āā".

5.2 Zeilenabstände bei Cambria Math

In Word 2000 habe ich Probleme mit dem Zeilenabstand bei der Schriftart Cambria Math. Die Zeilenabstände werden dabei viel zu groß.



Lösung: Feste Zeilenabstände verwenden:

Pfad: "Format → Absatz → Zeilenabstand: Genau"

6 NumLock-Taste deaktiviert sich in Word 2010 nach Anwendung von SendKeys-Befehlen

Bei der Ausführung einiger Makros stellte sich die immer wieder die NumLock-Taste ab, nachdem Windows 7 installiert worden war. Als Fehlerursache stellten sich zwei "SendKeys"-Befehle in einem Makro heraus. Ich hatte sie verwendet, um auf ein Menüband (Ribbon) in Word 2010 zuzugreifen, zum Beispiel:

```
SendKeys "(%r)" ' Imitiert Tastenfolge Alt + r für das Aktivieren des Ribbons "Start".
SendKeys "(%p)" ' Imitiert Tastenfolge Alt + p für das Aktivieren des Ribbons "Überprüfen".
```

Nachdem ich diese SendKey-Befehle entfernt hatte, trat das Problem nicht mehr auf.

Man kann natürlich einige Makrozeilen schreiben, welche die NumLock-Taste anschließend wieder aktivieren (siehe 6.2). Die sicherste Lösung für den Zugriff auf Menübänder ist die folgende von Tony Jollans (siehe 6.1).

6.1 aufwendige, aber zuverlässige Lösung, um eine Menüband-Registerkarte (Ribbon-Tab) in Word 2010 per VBA-Makro anzusprechen

Man kann die Ribbon-Tabs in Word 2010 mit einem sehr aufwendigen Makro ansteuern, siehe Artikel von Tony Jollans unter:

<http://www.wordarticles.com/Shorts/RibbonVBA/RibbonVBADemo.php>.

Ich musste es ein klein wenig modifizieren, um es für eine deutschsprachige Word-Version zu nutzen. Hier ein Beispiel, um die Registerkarte "Überprüfen" per Makro zu aktivieren:

Sub Ribbon_Überprüfen_in_Word_2010_aktivieren()

```
' getestet in Word 2010 mit Windows 7.
' In Word 2010 gibt es Ribbons statt Symbolleisten. Diese Ribbons können nicht mit VBA-Befehlen
' angesteuert werden.
```

```
Call Ribbon_aufrufen_per_VBA ("Überprüfen")
```

```
' und wahlweise folgende Zeile:
```

```
CommandBars("Reviewing").Controls(1).ListIndex = 1 ' "Abgeschlossen: Markups anzeigen"
```

```
End Sub
```

```
' *****
```

In einem separaten Makro-Modul ist das Folgende unterzubringen:

```
' aus: http://www.wordarticles.com/Shorts/RibbonVBA/RibbonVBADemo.php ,
```

```
' Download 10. Juni 2013, etwas modifiziert.
```

```
' Definitions and Procedures relating to Accessibility, used by the Ribbon VBA '
```

```
' Demonstration UserForm. The constants have been lifted from oleacc.h, and are '
```

```
' just a subset of those available. '
```

```
' Tony Jollans, August 2008. '
```

Option Explicit

```
Public Const CHILDID_SELF As Long = &H0&
```

```
Public Const STATE_SYSTEM_UNAVAILABLE As Long = &H1&
```

```
Public Const STATE_SYSTEM_INVISIBLE As Long = &H8000&
```

```
Public Const STATE_SYSTEM_SELECTED As Long = &H2&
```

```

Public Enum RoleNumber
    ROLE_SYSTEM_CLIENT = &HA&
    ROLE_SYSTEM_PANE = &H10&
    ROLE_SYSTEM_GROUPING = &H14&
    ROLE_SYSTEM_TOOLBAR = &H16&
    ROLE_SYSTEM_PAGETAB = &H25&
    ROLE_SYSTEM_PROPERTYPAGE = &H26&
    ROLE_SYSTEM_GRAPHIC = &H28&
    ROLE_SYSTEM_STATICTEXT = &H29&
    ROLE_SYSTEM_TEXT = &H2A&
    ROLE_SYSTEM_BUTTONDROPDOWNGRID = &H3A&
    ROLE_SYSTEM_PAGETABLIST = &H3C&

```

```

End Enum

```

```

Private Enum NavigationDirection
    NAVDIR_FIRSTCHILD = &H7&

```

```

End Enum

```

```

Private Declare Function AccessibleChildren _

```

```

    Lib "oleacc.dll" _
    (ByVal paccContainer As Object, _
    ByVal iChildStart As Long, _
    ByVal cChildren As Long, _
    rgvarChildren As Variant, _
    pcObtained As Long) _
    As Long

```

```

Public Function GetAccessible _

```

```

    (Element As IAccessible, _
    RoleWanted As RoleNumber, _
    NameWanted As String, _
    Optional GetClient As Boolean) _
    As IAccessible

```

```

' *****

```

```

' This procedure recursively searches the accessibility hierarchy, starting '
' with the element given, for an object matching the given name and role. '
' If requested, the Client object, assumed to be the first child, will be '
' returned instead of its parent.

```

```

' Called by: RibbonForm procedures to get parent objects as required '
'           Itself, recursively, to move down the hierarchy '
' Calls: GetChildren to, well, get children. '
'           Itself, recursively, to move down the hierarchy '
' .....

```

```

Dim ChildrenArray()
Dim Child As IAccessible
Dim ndxChild As Long
Dim ReturnElement As IAccessible

```

```

If Element.accRole(CHILDID_SELF) = RoleWanted _
And Element.accName(CHILDID_SELF) = NameWanted Then
    Set ReturnElement = Element
Else ' not found yet
    ChildrenArray = GetChildren(Element)
    If (Not ChildrenArray) <> True Then
        For ndxChild = LBound(ChildrenArray) To UBound(ChildrenArray)
            If TypeOf ChildrenArray(ndxChild) Is IAccessible Then
                Set Child = ChildrenArray(ndxChild)
                Set ReturnElement = GetAccessible(Child, _
                    RoleWanted, _
                    NameWanted)
            If Not ReturnElement Is Nothing Then Exit For

```

```

        End If ' Child is IAccessible
    Next ndxChild
    End If ' there are children
End If ' still looking
,
If GetClient Then
    Set ReturnElement = ReturnElement.accNavigate(NAVDIR_FIRSTCHILD, _
        CHILDID_SELF)

    End If
    Set GetAccessible = ReturnElement
End Function
'-----

```

Private Function GetChildren _

```

    (Element As IAccessible) _
    As Variant()
' General purpose subroutine to get an array of children of an IAccessible '
' object. The returned array is Variant because the elements may be either '
' IAccessible objects or simple (Long) elements, and the caller must treat '
' them appropriately.
,
' Called by: GetAccessible when searching for an Accessible element
' Calls: AccessibleChildren API
,
Const FirstChild As Long = 0&
,
Dim NumChildren As Long
Dim NumReturned As Long
Dim ChildrenArray()
,
NumChildren = Element.accChildCount
,
If NumChildren > 0 Then
    ReDim ChildrenArray(NumChildren - 1)
    AccessibleChildren Element, FirstChild, NumChildren, _
        ChildrenArray(0), NumReturned
    End If
    GetChildren = ChildrenArray
End Function
'-----

```

Public Sub Ribbon_aufrufen_per_VBA(TabName As String)

```

' aus: http://www.wordarticles.com/Shorts/RibbonVBA/RibbonVBADemo.php,
' Download 10. Juni 2013. modifiziert 11. Juni 2013 für deutsche Anwendung.
"Sub SwitchTab "Review"
"Dim TabName As String
Dim RibbonTab As IAccessible
Set RibbonTab = GetAccessible(CommandBars("Ribbon"), _
    ROLE_SYSTEM_PAGETAB, _
    TabName)
If Not RibbonTab Is Nothing Then
    If ((RibbonTab.accState(CHILDID_SELF) _
        And (STATE_SYSTEM_UNAVAILABLE Or STATE_SYSTEM_INVISIBLE)) = 0) Then
        RibbonTab.accDoDefaultAction CHILDID_SELF ' der endgültige Aufruf des Ribbons.
    End If
End If
End Sub
'-----

```

6.2 CapsLock wieder-ausschalten und NumLock wieder-einschalten

Die folgende Prozedur mit API-Funktionen setzte ich aus mehreren Quellen zusammen, die ich im Internet fand.

```
" ((oben im Modul einfügen)):
' API-Funktion für Numlocktaste aktivieren und CAPSLock deaktivieren
Private Declare Function GetKeyState Lib "user32" ( _
    ByVal nVirtKey As Long) As Integer

Private Declare Sub keybd_event Lib "user32" ( _
    ByVal bVk As Byte, _
    ByVal bScan As Byte, _
    ByVal dwFlags As Long, _
    ByVal dwExtraInfo As Long)

Sub Capslock_aus_Numlock_ein()
' Stellt CapsLock-Taste aus und NumLock-Taste ein.
' Lösung aus dem Internet, stark bearbeitet.
' NUM-Lock aktivieren (falls deaktiviert):
' vbKeyNumlock = 144
If GetKeyState(vbKeyNumlock) <> 1 Then
    keybd_event vbKeyNumlock, 1, 0, 0
    keybd_event vbKeyNumlock, 1, 2, 0
End If
' Caps-Lock deaktivieren (falls aktiviert):
' vbKeyCapital = 20
If GetKeyState(vbKeyCapital) <> 0 Then
    keybd_event vbKeyCapital, 1, 0, 0
    keybd_event vbKeyCapital, 1, 2, 0
End If
End Sub
'-----
```

7 Probleme mit **Windows 7**

7.1 Problem mit sich drehendem Bildschirm in Windows 7

Wenn man Befehlen oder Makros Tastenkürzel zugewiesen hat, funktionieren nicht mehr alle unter Windows 7. Zum Beispiel bei "**AltGr** + **Pfeil-rechts**" dreht sich der Bildschirm.

Abhilfe: Dann kann man mit "**AltGr** + **Pfeil nach oben**" den Bildschirm zurückdrehen.

7.2 Das Makro für den Briefkopf mit aktuellem Datum "Sub AutoNew()" funktioniert unter Windows 7 nicht mehr zuverlässig

Ich benutze die AutoNew-Prozedur, um einen Briefkopf mit Name und aktuellem Datum zu erzeugen. Unter Windows XP läuft die Prozedur einwandfrei von Word 2000 bis Word 2010.

Problem: Unter Windows 7 reagiert die Prozedur "Sub AutoNew()" nicht, wenn noch keine Word-Datei geöffnet ist. Beim Start von Word unter Windows 7 wird nur die Prozedur "Sub AutoExec()" ausgeführt. Ist aber bereits eine Word-Datei geöffnet, reagiert die Prozedur "Sub AutoNew()" ganz normal wie bisher unter Windows XP auf das Erzeugen einer neuen Datei. Es wird dann aber nicht die Prozedur "Sub AutoExec()" ausgeführt.

Lösungsversuch: Würde man in der "Sub AutoExec()" die Routine für Name und Datum von Briefköpfen unterbringen, gibt es eine Fehlermeldung, denn solange "Sub AutoExec()" läuft, ist in Windows 7 noch keine einzige Datei geöffnet. Selbst eine Warteschleife, die mehrere Sekunden wartet, würde hier nicht helfen. Sogar eine MsgBox in der AutoExec, die ich erst nach mehreren Sekunden schloss, half hier nicht!



Bild 7.1: Screenshots der Fehlermeldung bei AutoExec

Der VBA-Befehl `Application.Windows.Count` ergab in der "Sub AutoExec()" stets nur "0".

Lösung: Mittels des "OnTime"-Befehl wird mit Verzögerung (1 Sekunde reicht) ein weiteres Makro gestartet, welches "Sub AutoNew()" aufruft, wenn noch keine Word-Datei geöffnet bzw. erzeugt worden ist. Erst mit einer Zeitverzögerung, nach dem Beenden der Prozedur "Sub AutoExec()", kann ich jetzt prüfen, ob es eine geöffnete Word-Datei gibt, und ggf. eine neue Datei erzeugen, wobei nun "Sub AutoNew()" funktioniert, weil inzwischen eine neue Datei erzeugt worden ist.

Wenn ich statt "Public Sub" für die DelayProzedur "Private Sub" schreibe, funktioniert sie unzuverlässig.

In der "Sub AutoExec()" habe ich vor der Warteschleife noch weitere Befehle für andere Zwecke untergebracht. Möglicherweise ist diese Warteschleife überflüssig, wenn die Prozedur außer dem OnTime-Befehl nichts Weiteres enthält.

Die Prozedur "Sub AutoNew" enthält einige Befehlszeilen zur Erzeugung eines Briefkopfs, wenn die Datei noch leer ist und auf der Normal.dot beruht.

```

Sub AutoExec()
  Dim tt As Single
  tt = Timer
  Do While Timer < tt + 0.2 'Sekunden
    DoEvents
  Loop
  ' ***
  "If Application.Windows.Count = 0 Then Application.Documents.Add ' ergibt hier immer 0.
  Application.OnTime When:=Now + TimeValue("00:00:01"),
  Name:="DelayProzedur_für_Neue_Datei"
End Sub

Public Sub DelayProzedur_für_Neue_Datei()
  ' 10. September 2013, 4. Mai 2016.
  ' Makro mit verzögertem Start für AutoNew, wenn wirklich neues Dokument.
  "MsgBox "Hallo!" & Application.Windows.Count
  If Application.Windows.Count = 0 Then ' gibt Fehler bei geschützt geöffneten Dateien
  " Application.Documents.Add
  Else
  Call AutoNew
  End If
End Sub

Sub AutoNew()
  ' Dipl.-Ing. H. Chris Gast, 26. März 2014.
  ' In jeder neuen Datei sind in der ersten Zeile Name + Datum einzufügen.
  ' Abfrage, ob neue Datei, die auf der Normal.dot basiert:
  If ActiveDocument.AttachedTemplate <> Application.NormalTemplate _
  Or Len(ActiveDocument.Range) > 1 Then Exit Sub
  ' *****
  ' Name und Datum:
  Selection.TypeText Text:="GEWÜNSCHTER NAME" & vbTab
  Selection.InsertDateTime DateTimeFormat:="dddd, d. MMMM yyyy", _
  InsertAsField:=False, DateLanguage:=wdGerman, _
  CalendarType:=wdCalendarWestern, InsertAsFullWidth:=False
  Selection.TypeParagraph
  Selection.TypeParagraph
End Sub

```

Der "GEWÜNSCHTER NAME" ist nach Bedarf zu ergänzen.

7.3 Verlust des Inhalts der Zwischenablage beim Erzeugen einer neuen Datei

Seit Juli 2015 verschwindet bei mir der Inhalt der Zwischenablage beim Erzeugen einer neuen Datei. Im Herbst 2015 war es wieder gut, aber im November oder Dezember 2015 tauchte das Problem nach einem Update von Windows oder Office erneut auf.

Die im Internet genannten Lösungen funktionierten bei mir nicht (Löschung eines störenden neuen Add-Ins in Word, Pfad: Datei → Optionen → AddIns.

Ich stellte fest, dass bei mir der Inhalt der Zwischenablage während des Makros für den Briefkopf mit Namen und Datum sowie Seitenzahlen-Generierung verschwand.

Meine Abhilfe (umständlich, aber es funktioniert):

Ich deaktiviere zuerst alle Automakros in der "Sub AutoNew()":

```
Sub AutoNew()  
WordBasic.DisableAutoMacros 1  
"... usw.
```

Dann erzeuge ich eine unsichtbare temporäre Datei,

```
Set Dok2 = Documents.Add(DocumentType:=wdNewBlankDocument, Visible:=False)
```

in die ich den Inhalt der Zwischenablage (Clipboard) einfüge.

Nun erzeuge ich die endgültige neue Datei mit Briefkopf und Seitenzahlen.

Zuletzt kopiere ich aus der unsichtbaren Datei den Inhalt wieder in die Zwischenablage (Clipboard) und lösche dann die unsichtbare temporäre Datei.

8 Problem mit dem Entlinken von Feldern in Tabellen

Bei **Strg** + **Shift** + **F9** werden in einer Tabelle nicht alle Felder "entlinkt", und/oder es wird eine Tabellenzeilenanzahl zerstört.

Abhilfe bietet folgendes Makro, das ich mal im Internet fand und für mich anpasste:

Sub s_Unlink()

' In Tabellen über mehrere Tabellenzellen funktioniert der direkte Unlink-Befehl nicht.

'' Ferner muss zum Entlinken einer automatischen Nummerierung rückwärts vorgegangen werden!

Dim Range1 **As** Range, i **As Integer**

Set Range1 = Selection.Range

' Rückwärts zählen und unlinken, sonst wird eine vorhandene Nummerierung

*' wie in {AUTONUM * Arabic \s " } zerstört!*

For i = Range1.Fields.Count **To** 1 **Step** -1

 Range1.Fields(i).Unlink

Next i

End Sub

Ich habe diesem Makro dann das Tastenkürzel **Strg** + **Shift** + **F9** zugewiesen. Dann tritt es an die Stelle des Befehls.

9 Einstellungen von Word **sichern** beim Übertragen der persönlichen Word-Einstellungen auf einen neuen Computer

9.1 Wo ist die **Normal.dotm** versteckt, und wo ist der StartUp-Ordner?

Mit "Suchen" fand ich im neuen Computer mit Windows 7 die "Normal.dotm" nicht, auch nicht, wenn ich im Explorer die Einstellung wählte "*ausgeblendete Dateien...anzeigen*".

Abhilfe: Öffne eine Word-Datei,

Drücke **Alt** + **F11**,

Dann öffnet sich im Makro-Explorer das "Direktfenster".

Gebe ein:

```
Debug.Print Options.DefaultFilePath(wdUserTemplatesPath)
```

Dann erhält man in der Zeile darunter einen Pfad wie:

```
c:\users\...\microsoft\vorlagen
```

Diesen Pfad kopiert man mit **Strg** + **C** und gibt es im Explorer ganz oben in der Leiste ein (in Windows 7 diese Leiste erst anklicken, dass sie von grau in weiß wechselt).

Genauso findet man den **Word-StartUp-Ordner**:

```
Debug.Print Options.DefaultFilePath(wdStartupPath)
```

Das ergibt einen Pfad wie:

```
c:\users\...\microsoft\word\startup
```

Dort kann man zum Beispiel meine fertige Dokumentvorlage mit Makros nach [3] unterbringen.

9.2 Bilder-Einfüge-Pfad kann nicht mehr voreingestellt werden

Beim Einfügen von Bildern mit dem Tastenkürzel "**Alt** + **E**, **G**, **A**" (oder über Menü) wird im Explorer der Ordner "Meine Bilder" geöffnet. Dies lässt sich in **Word 2010** nicht mehr verstellen. Im Direktfenster des Makro-Explorers (unter **Alt** + **F11**) mit

```
Debug.Print Application.Options.DefaultFilePath(wdPicturesPath)
```

erhält man fälschlich:

```
c:\program files\microsoft office\office14
```

Gefundene Erklärung:

"Application.Options.DefaultFilePath(wdPicturesPath)" wird bei "Word 2010" nicht gebraucht und ist standardmäßig leer. Das Objektmodell gibt den Pfad zu den Office-Anwendungen zurück.

Abhilfe (Lösung):

Es wird zuerst der gewünschte Pfad per Hand aus dem Explorer ausgelesen, zum Beispiel bei mir zuhause ist dieser:

```
D:\transfer
```

Dann wird **mittels des folgenden Makrobefehls (VBA)** dieser Pfad den Word-Einstellungen als Option zugewiesen. Glücklicherweise ist dieser Makrobefehl in Word 2010 noch gültig. Im Direktfenster (Aufrufen mit **Alt** + **F11**) gibt man ein (hier mein Beispiel):

```
Application.Options.DefaultFilePath(wdPicturesPath) = "D:\transfer"
```

Jetzt ergibt der obige Befehl:

```
Debug.Print Application.Options.DefaultFilePath(wdPicturesPath)
```

wie gewünscht: d:\transfer

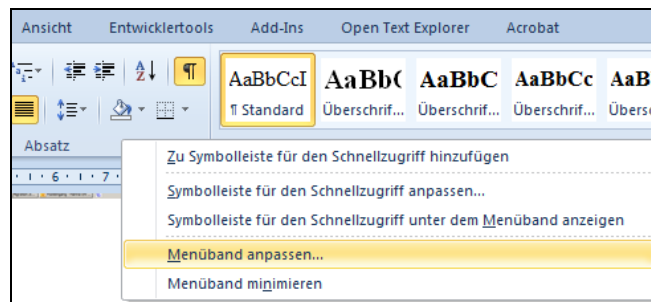
Das Schöne daran ist:

Diese Einstellung bleibt nach Runterfahren des Computers erhalten. Man braucht sie nicht bei jedem Hochfahren des Computers zu wiederholen.

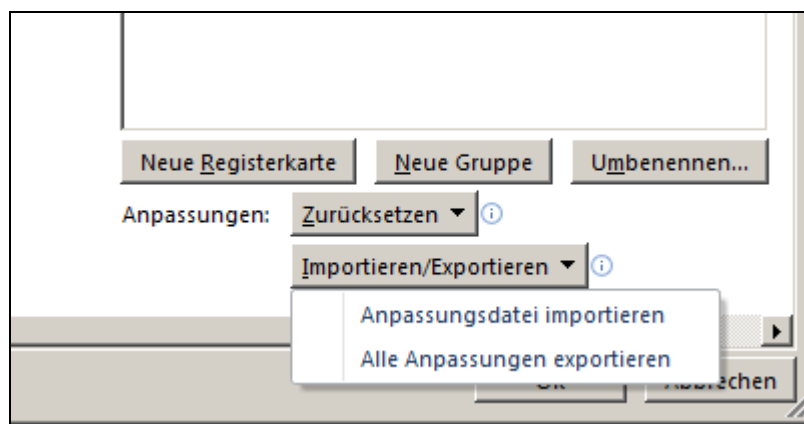
9.3 Per Hand zu übertragende Einstellungen

Folgendes ist zu speichern:

1. Die "Normal.dot" bzw. "Normal.dotm" auf einen USB-Stick, eine mobile Festplatte oder auf ein Netzlaufwerk kopieren und im neuen Computer anstelle der dort neu erzeugten leeren Normal.dotm einfügen. Darin sind die Formatvorlagen und ggf. Makros enthalten, sowie ggf. die AutoKorrekturen mit "Formatierten Text".
2. Bis Word 2003 werden die Symbolleisten in der Normal.dot gespeichert.⁶ Dagegen befinden sich in Word 2010 die **Ribbon-Einstellungen** (Schnellstartleiste und Menüleiste) an einem anderen Ort. Sie müssen vor dem Computer-Wechsel exportiert werden:
 - a) Rechtsklick in das Menüband (Ribbon);
 - b) "Menüband anpassen" wählen;



- c) Unten rechts im erscheinenden Feld auswählen: "Importieren/Exportieren", dann "Alle Anpassungen exportieren", externes Verzeichnis wählen (USB-Stick usw.), dann "OK".



- d) Die neu entstandene Datei heißt: "Word-Anpassungen.exportedUI"
- e) im neuen bzw. neu aufgesetzten Computer entsprechend "Anpassungsdatei importieren" wählen.

⁶ Übertragung von Symbolleisten bis Word 2003 ähnlich wie Makromodule mit Alt+F8, dann "Organisieren".

3. Für die **AutoKorrekturen** (ohne "Formatierten Text") suche man nach den ".acl"-Dateien.

In Windows XP sind sie in der Regel unter:

C:\Dokumente und Einstellungen\Namenskürzel\Anwendungsdaten\Microsoft\Office

In Windows 7 sind sie in der Regel unter:

C:\Users\Namenskürzel\AppData\Roaming\Microsoft\Office

Dieser Pfad ist oben im Explorer einzugeben. Mit der Suchfunktion im Explorer fand ich sie nicht! Notfalls hilft hier das Makro in B.2.

Folgende ".acl-Dateien" sind von Bedeutung:

MSO1031.acl für Deutsch (Deutschland)

MSO1033.acl für Englisch (USA)

MSO2057.acl für Englisch (Großbritannien)

MSO1036.acl für Französisch (Frankreich)

Word erzeugt bei automatischer Spracherkennung fälschlich auch AutoKorrekturen für weitere Sprachen bis hin zu "MSO3179.acl" für Quechua (Peru). Diese braucht man natürlich nicht zu übertragen.

Die formatierten AutoKorrekturen sind in der Normal.dot bzw. Normal.dotm gespeichert. Hierfür ist entweder diese Normal.dot in den neuen Computer zu übertragen oder das Formblatt mit Makro nach[4] zu verwenden.

4. **Das Benutzerwörterbuch.** Der Name der Wörterbuch-Dateien endet auf ".dic", meist "Custom.dic".⁷

Die Pfade der Benutzerwörterbücher findet man in Word 2010 unter

"Datei → Optionen → Dokumentprüfung → Benutzerwörterbücher".⁸

Dort kopiert man sich dann jeweils aus dem grauen Feld den jeweiligen "Dateipfad" heraus und gibt ihn im Explorer oben in der Adressleiste ein.

Meist bis Word 2003: C:\Dokumente und Einstellungen\...\Anwendungsdaten\Microsoft\Proof.

bzw. in Word 2010: C:\Users\...\AppData\Roaming\Microsoft\UProof

Die Benutzerwörterbuch-Dateien werden dann auf einem Netzlaufwerk oder auf einem USB-Stick gespeichert. Ich empfehle bei dieser Gelegenheit eine Nachbearbeitung von Hand (mit dem Editor). Beim Übergang von alten Word-Versionen auf Word 2010 ist dabei statt der bis Word 2003 üblichen Codierung "ANSI" die Codierung "Unicode" zu wählen.

Das Benutzerwörterbuch bzw die Benutzerwörterbücher sind im neuen Computer bei Word 2010 im Ordner "C:\Users\...\AppData\Roaming\Microsoft\UProof" zu speichern, ggf. nur den Inhalt dort ergänzen, wenn dort die "Custom.dic" bereits wichtige Einträge enthält.

⁷ Die Suche nach ".dic" im Explorer funktionierte bei mir nicht. Eventuell müsste man die Ansichtsoptionen ändern und/oder Administratorenrechte haben.

⁸ Bis Word 2003 unter "Extras → Optionen" usw.

9.4 **Formblatt mit Makros** zur Übertragung der Tastenkürzel und der (deutschen) AutoKorrekturen auf einen neuen Computer

Mittels eines Formblatts mit Makros können eigene Tastaturkombinationen (Tastenkürzel) und AutoKorrekturen auf einen neuen Computer oder auf eine neue Word-Version übertragen werden (erprobt von Word 2000 bis Word 2010).

Das **Formblatt mit Anknickbuttons** gibt es als Download unter:

http://www.siebener-kurier.de/chris-aufsaeetze/Formblatt-Tastenkuerzel+Autokorrekturen_uebertragen.doc, siehe [5].

Das Formblatt ermöglicht im oberen Bereich die Übertragung von Tastenkürzeln und im unteren Bereich die Übertragung der formatierten und der deutschen "AutoKorrekturen". Seit Januar 2015 können damit auch AutoKorrekturen übertragen werden, die Tabellen oder Absatzmarken enthalten.

9.4.1 Beschreibung der Funktion "Tastenkürzel übertragen":

Mit der ersten Prozedur im Formblatt "*eigene_Tastenkürzel_auflisten_tabellarisch*" werden alle eigenen Tastenkürzel tabellarisch aufgelistet. Es entsteht dabei an der Cursorstelle ein Tabelle (hier ein Auszug aus meinem Computer mit Word 2000):

Tastenkürzel	Makro- bzw. Befehlsname oder Sonderzeichen	Befehls-kategorie
Strg+\	Arial	3
Strg+Umschalt+G	InsertPicture	1
Strg+Umschalt+P	Normal.NewMacros.Druckern_sparsam_2Seiten_pro_Blatt	2
Strg+Umschalt+S	FileSaveAs	1
Alt+4	Überschrift 4	5
Alt+S	Standard	3
Alt+V	Connect	1
Alt+Umschalt+C	Normal.Unicode_Analyse.Hexcode_und_Name_ermitteln	2
Alt+Strg+C	ç	6
Alt+Strg+W	∞	6
Alt+Strg+Umschalt+R	Normal.NewMacros.Farbmarkierung_rot_suchen	2

Die Befehlskategorie braucht das VBA-Word-Makro, um später in einem neuen Computer unter Word die Tastenkürzel neu zuzuweisen. "2" steht für Makros, "6" für Sonderzeichen einfügen, "1" für Befehl wie "Bild einfügen", "3" für Schriftart oder Formatvorlage zuweisen usw.

Die entstehende Datei mit der Tabelle der Tastenkürzeln enthält jetzt auch oben auf der 1. Seite das Datum und Computer-Name.

Mit der zweiten Prozedur (*Befehl_auf_Taste legen*) wird im Zielcomputer die Tabelle zeilenweise ausgelesen und deren Tastenzuordnungen in die Normal.dot einprogrammiert.

9.4.2 Kurzbeschreibung zum Übertragung von AutoKorrekturen

Man kann alle AutoKorrekturen ohne Formatierungen (Einträge ohne Sonderzeichen, ohne Formate, ohne Bilder) bei allen Word-Versionen übertragen, indem man einfach die "*.acl"-Dateien überträgt.

Dagegen die formatierten Einträge werden in allen Word-Versionen in der Normal.dot (bzw. Normal.dotx) gespeichert. Hier muss weiterhin ein Makro verwendet werden.

Das fertige Formblatt hilft, die ".acl"-Dateien Dateien zu finden und zu unterscheiden. Das Makro im Formblatt Stand Januar 2015 überträgt jetzt nur noch die deutschen und die formatierten AutoKorrekturen in einen neuen Computer.

Die ursprünglichen Makros zur Übertragung von Tastenkürzeln übernahm ich aus der "Support.dot", aber es zeigte sich, dass die Erstellung einer Tabelle der AutoKorrektur-Einträge fehlerhaft wird, wenn Absatzmarken oder Tabellen enthalten sind. Dazu musste ich das Makro ändern.

Die entstehende Datei mit der Tabelle der AutoKorrekturen enthält jetzt (Jan. 2015) auch oben auf der 1. Seite das Datum und Computer-Name.

Backup aus Computer: PCBE1921 → Speicherdatum: 03.02.2015

AutoCorrect-Backup-Document

Name	Value	Formatiert
(c)	©	Falsch
(e)	€	Falsch
(r)	®	Falsch
(tm)	™	Falsch
...	...	Falsch
arbeiten	arbeiten	Falsch
Abmachung	Abmachung	Falsch
Accessoir	Accessoire	Falsch
Ackessoir	Accessoire	Falsch
adnere	andere	Falsch

Die fremdsprachigen AutoKorrekturen werden von meinem Formblatt (Stand Januar 2015) nicht erfasst, es wird aber unter der Tabelle mit den AutoKorrekturen der Pfad zu den ".acl"-Dateien aufgezeigt und diese Dateien erklärt.

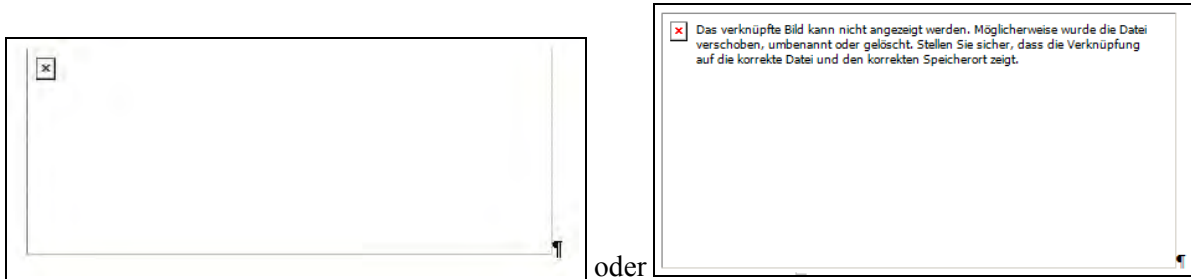
Ferner wird jetzt bei umfangreichen AutoKorrektur-Listen beim Übertragen in den neuen Computer abgefragt, ob die dort schon vorhandenen (deutschen) AutoKorrekturen gelöscht werden sollen.

Zu Problemen mit AutoKorrekturen siehe Abschnitt 3.

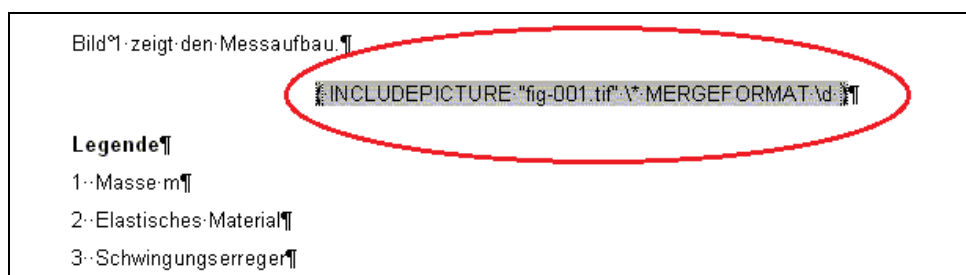
10 Probleme mit falsch verlinkten Bildern

10.1 Fehlende Bilder

Wenn Bilder nicht erscheinen, obwohl sie eigentlich richtig verlinkt sind, gibt es folgende Möglichkeiten.



- a) Sind die Bild-Dateien in dem festgelegten Ordner vorhanden?
- b) Bis Version Word 2003 und in Word 2010 in "kompatiblen Dateien":
- 1) das ganze Dokument markieren mittels Drücken von **Strg**+**A**
 - 2) alle Felder aktualisieren durch das Drücken von **F9**. Bei Abfrage "OK" anklicken.
Bildverknüpfungen sind hier normale Feldfunktionen.
- b) In Word 2010 ohne Kompatibilitätsmodus, oder in Word, wenn das obige nicht ausreicht, empfehle ich, folgendes zu versuchen:
1. **Unter Word 2000/Word 2003:** "Bearbeiten → Verknüpfungen";
 2. **Unter Word 2010:** "Datei → Informationen", dort rechts unten → "Verknüpfungen mit Dateien bearbeiten";
Dort jeweils anklicken: "**Quelle ändern**" und das richtige Bild auswählen.
Oft ist nur der Pfad falsch.
- c) Wenn alle Bilder systematisch mit einer einheitlich-falschen Pfadangabe verlinkt sind, kann man (bis Word 2003, in Word 2010 nur im Kompatibilitätsmodus) einfacher mit **Alt** + **F9** umschalten auf "Feldfunktion anzeigen" (in der ganzen Datei) und mit Suchen/Ersetzen alle falschen Pfade **auf einmal** beseitigen, so dass die Verknüpfung folgt aussieht:



Anschließend mit **Alt** + **F9** die Feldfunktionsanzeige wieder abschalten und wie oben in Punkt a) alle **Verknüpfungen aktualisieren** (**Strg**+**A**, dann **F9**).

Bei ".docx"-Dateien ohne Kompatibilität funktioniert Punkt c) nicht mehr. Bildverknüpfungen sind dort meistens keine normale Feldfunktionen, die man mit **Alt** + **F9** sichtbar machen kann. In der Regel **funktioniert jedoch die Bildanzeige trotzdem**, wenn sich die Bilder alle vollständig in demselben Ordner wie die Word-Datei befinden (*sind dort auch wirklich die richtigen Bilder?*), unabhängig davon, wo sich die Bild-Dateien beim Schreiben der Word-Datei ursprünglich befanden. Aber wehe, anders aussehende Bilder, jedoch mit gleichem Dateinamen befinden sich ebenfalls irgendwo auf der Festplatte: Das kann gelegentlich zu Chaos führen.

10.2 Bei Word 2010 herkömmliche Feldfunktion erzwingen

Wenn man unbedingt will, kann man auch in Word 2010 bei nicht mit Word 2003-kompatiblen Dateien verknüpfte Grafiken mit funktionierender Feldfunktion erzeugen. Dazu muss erst eine Feldfunktion erzeugt werden, und darin "IncludePicture" gewählt werden:

Pfad: Einfügen → Schnellbausteine → Feld...

Dort wählt man im Dialog "Feldnamen:" (durch Scrollen) "INCLUDEPICTURE"

Unter "Dateiname oder URL" muss der exakte Dateiname der Grafik eingetragen werden (eventuell mit Pfad). Rechts sollte als weitere "Feldoptionen" festgelegt werden: "Datei nicht im Dokument speichern", was den Schalter "\d" erzeugt.

Die Feldfunktion sieht dann folgendermaßen aus: `{ INCLUDEPICTURE "Testbild.tif" * MERGEFORMAT \d }`

Aber kaum verschiebt man den Absatz, verwandelt sich die Feldfunktion in die neuartige, schwer zu bearbeitende Verknüpfung, an die man mit **Alt** + **F9** nicht mehr herankommt. Von daher rate ich davon ab. Weiteres siehe den Artikel von Holger Matthes: [9].

10.3 Bilder der falschen Datei bei Word 2010

Bei nicht mit Word 2003-kompatiblen Dateien kann es (in Word 2010) zu fehlerhaften Bildverknüpfungen kommen, wenn mehrere Word-Dateien geöffnet sind, und die Dateinamen der Bilder gleich sind. Dies passiert auch, wenn diese korrekt mit relativen Pfaden verknüpft worden waren.

Mir wurde gesagt, dass in Windows 7 auf der Festplatte die Ordner-Pfade nur virtuell sind. Der Pfad zu "Bild_001.tif" usw. ist also kein echter Pfad. Und wenn dann eine andere Word-Datei auch ein "Bild_001.tif" enthält, und in einer Word-Sitzung beide Word-Dateien geöffnet wurden (zeitgleich oder nacheinander), dann kann es in Word 2010 bei den Bildern Chaos geben. Möglicherweise spielen hier auch temporäre Wiederherstellungsdateien eine Rolle.

Abhilfe:

- Grundsätzlich niemals zwei verschiedene Bild-Dateien im Computer mit demselben Dateinamen versehen;
- Nicht mehrere Word-Dateien mit Bildverknüpfungen in derselben Word-Sitzung aufmachen;
- Bilder fest einbinden, solange die Datei nicht zu groß wird.

Kleiner Vorteil des neuen "Systems":

Bilder werden angezeigt, wenn sich die Bild-Dateien in demselben Ordner befinden wie die Word-Datei, auch wenn beim Einlinken der Bilder diese sich ursprünglich in einem anderen Ordner befanden.

Kleiner Trost: Falsche Verknüpfungen lassen sich über

"Datei → Informationen → Verknüpfungen mit Dateien bearbeiten"
ändern, wenn auch mühsam, siehe folgenden Screenshot.

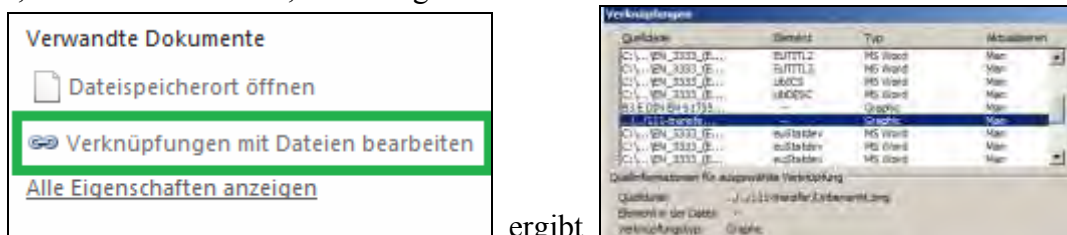


Bild 10.1: Screenshot aus "Datei → Informationen"

11 Formeln mit dem Formel-Editor OMML in Word 2010

Word 2010 (bzw. schon Word 2007) enthält gegenüber Word 2003 und älteren Versionen einen neuen integrierten "Formeleditor" ("Office Math Markup-Language", OMML)[10].

Wichtige Voraussetzung für den Formeleditor von Word 2010:

".docx"-Datei **ohne** Kompatibilität, sonst ist der Umfang zu gering.

Jedoch sind die neuen Formeln nicht rückwärtskompatibel. Beim Speichern als ".doc" oder beim Bearbeiten der ".docx"-Datei mit Word 2000 oder Word 2003 werden die neuen Formeln zu Grafiken.

Der alte Formeleditor 3.0 ist im neuen Word 2010 **inzwischen nicht mehr enthalten**. Er wurde mit den Updates vom Januar 2018 entfernt.

Lösungen für die Konversion von Formeln unterschiedlicher Formel-Editoren:

- MathType-Konverter, muss gekauft werden;
- Konvertieren von ".doc" über **OpenOffice** bzw. **LibreOffice** mit ".odt" und dann in ".docx";
- Weitere Angebote aus dem Internet (nicht getestet).

11.1 Aufruf des Formeleditor per Button oder per Makrobefehl

Wenn wider Erwarten im Menüband unter "Einfügen" in der Gruppe "Symbole" kein Button vorhanden ist, kann man sich diesen Button selbst einrichten. In das Menüband ("Ribbon") von Word 2010 oben am Bildschirm mit der rechten Maustaste klicken und "**Menüband anpassen...**" wählen, siehe auch Anhang C. Der Befehl findet man unter "Befehle auswählen" am einfachsten in der Einstellung "Alle Registerkarten → Einfügen → Symbole → Formel", siehe den folgenden Screenshot.

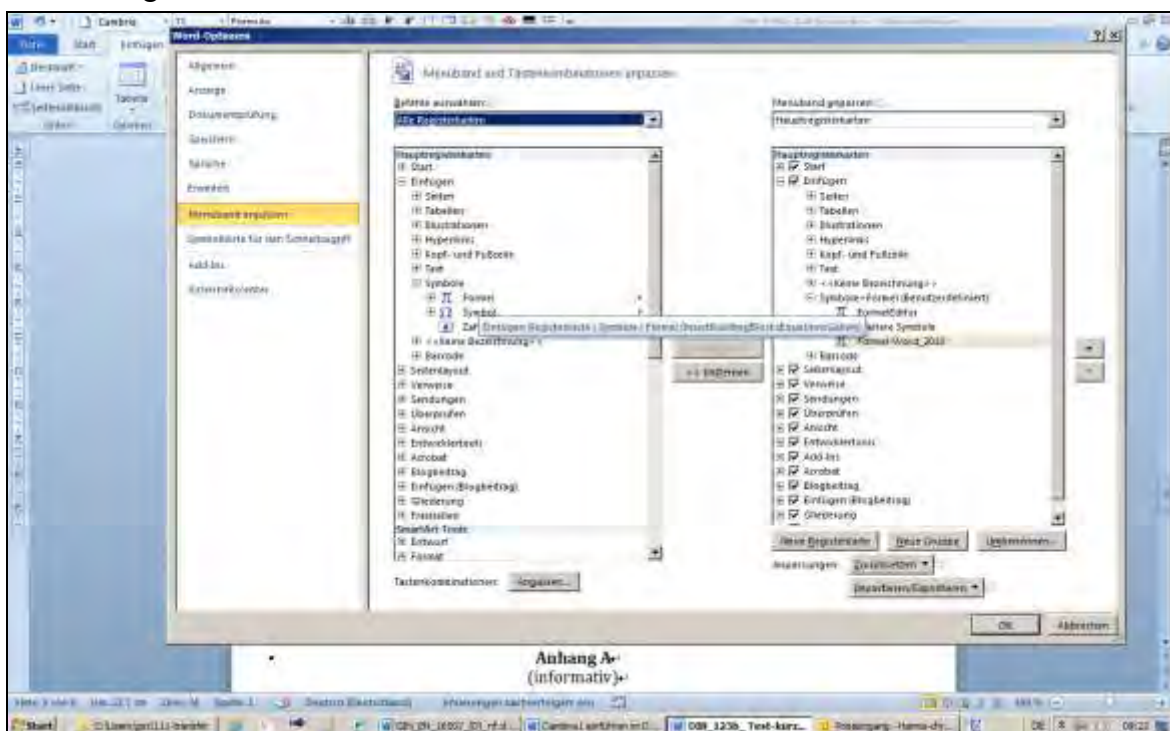


Bild 11.1: Einrichten eines Buttons im Menüband für den neuen Formeleditor

Aufruf des Formeleditors in Word 2010 per Makro :

Achtung: Markierter Text wird in die Formel eingefügt, inklusive markierter Absatzmarken. Deshalb hier sicherheitshalber "Selection.Collapse".

Sub Formeln_einfuegen_in_Word_2010()

' Enthält die Markierung Text, würde dieser in die Formel eingefügt (inklusive Absatzmarken):

Selection.Collapse

' *****

Selection.OMaths.Add Selection.Range

' Wahlweise Linksbündig setzen, wenn möglich:

On Error Resume Next

Selection.OMaths(1).ParentOMath.Justification = wdOMathJcLeft

On Error GoTo 0

End Sub

Problem bei einer neu erzeugten Datei:

Wenn ich eine neue Datei erzeuge, muss ich diese erst als ".docx" **ohne** Kompatibilität speichern, also keinen Haken im folgenden Screenshot.

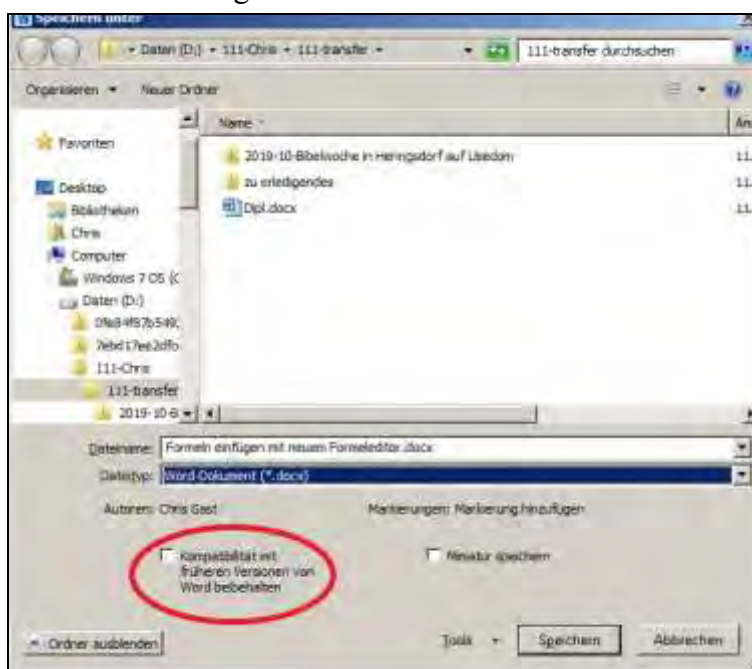


Bild 11.2: Speichern ohne Kompatibilität

11.2 Die Anwendung des Formeleditors in Word 2010

Neue Formel (bzw. Gleichung) erzeugen durch Anklicken des Buttons für Formeln in der Menüband-Registerkarte (Ribbon-Tab) "Einfügen".

Achtung: Anders als beim bisherigen Formeleditor 3 werden beim Aufrufen des Formeleditors für Word 2010 der Inhalt der markierten Stelle in die Formel eingefügt, inklusive Absatzmarken!

11.2.1 Gruppe Symbole

Beim Bearbeiten einer Formel den Cursor in die Formel stellen und oben weit rechts auf dem Bildschirm den Ribbon-Tab (das Menüband) "Formeltools, Entwurf" anklicken. Das folgende Bild zeigt die Grundansicht dieses Menübandes mit den drei Gruppen "Tools", "Symbole" und "Strukturen", siehe Bild.



Bild 11.3: Screenshot, Grundansicht des neuen Formeleditors von Word 2010

Bei der Gruppe "Symbole" kann man durch Klicken auf den kleinen Pfeil nach unten (rot markiert im obigen Bild) die Auswahl an Zeichen vergrößern (siehe folgendes Bild).

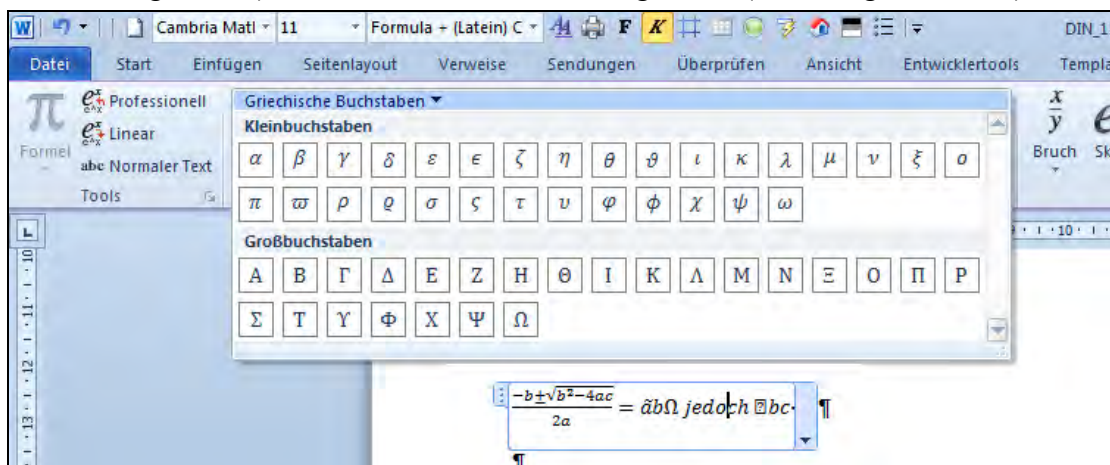


Bild 11.4: Screenshot, Formeleditor von Word 2010, Gruppe "Symbole"

Man kann auch eine andere Zeichengruppe wählen, wenn man auf das Dreieck rechts von "Griechische Buchstaben" im obigen Bild klickt, siehe folgendes Bild.

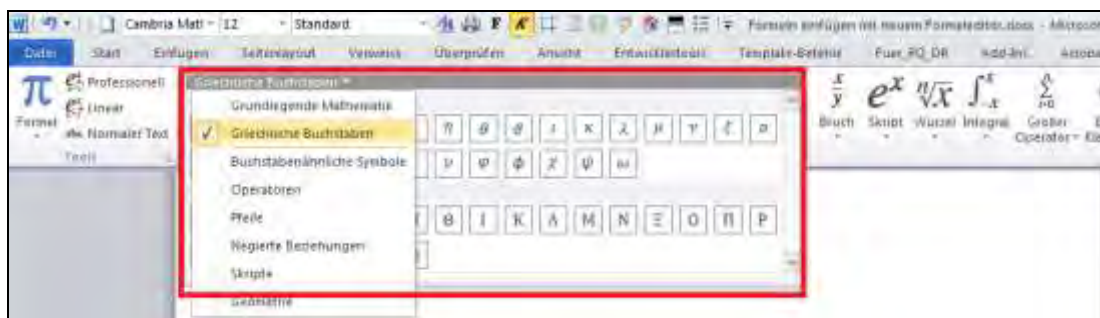


Bild 11.5: Gruppe "Symbole", größere Auswahl

Alle Schrift- und Sonderzeichen in den Formeln sind in Cambria bzw. in **Cambria Math**. Man kann auch über die Zwischenablage weitere Zeichen einfügen. Dabei geht jedoch die ursprüngliche Schriftart verloren. Für andere Schriftarten siehe Abschnitt 11.2.4 über die Gruppe "Tools".

Die Auswahl an Zeichen in Cambria bzw. Cambria Math ist im Prinzip riesig. Für eine noch größere Auswahl muss man statt dem oben gezeigten Weg direkt über die Registerkarte (Ribbon-Tab) "Einfügen" ein Zeichen auswählen, zum Beispiel:

"Einfügen → Symbole → Weitere Symbole,
"normaler Text", Subset "Griechisch und Koptisch".

Damit bekomme ich wie beim Schreiben eines normalen Textes alle gewünschten Zeichen:

$$x_1 = \frac{\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega}{\text{A B G D E Z H \theta I K \lambda M N \xi O P P E T Y \phi X \Psi \Omega}}$$

Auf diese Weise erhält man aber auch Zeichen, die der Formeleditor so nicht vorgesehen hatte:

$$x_2 = \varphi \phi \Phi \phi \text{f}$$

11.2.2 Zeichenformat

Über das Ribbon-Tab "Start" lassen sich einzelne Buchstaben in "steil" ändern und/oder in "Fett". Es gelang mir aber nicht gleich auf Anheiß, Ziffern kursiv zu setzen; dazu muss erst im Ribbon-Tab "Formeltools" links in der Gruppe "Tools" der Button "Normaler Text" angeklickt werden.

Hoch- und Tiefstellung funktioniert mittels der Gruppe Strukturen. Dort "Skript" anklicken, und dann die passende Hoch- oder Tiefstellung auswählen.

Die Schriftgröße richtet sich nach der Formatvorlage des Absatzes, in welchem die Formel oder Gleichung eingefügt wird. Sie kann geändert werden, wirkt sich dann stets auf die **gesamte** Gleichung aus.

11.2.3 Gruppe Struktur

Für die Gestaltung von Bruchstrichen, Integralen, Wurzelzeichen usw. wählt man in der Gruppe "Struktur" eines der Pull-Down-Menüs, im folgenden Bild zum Beispiel eines der Summenzeichen usw.

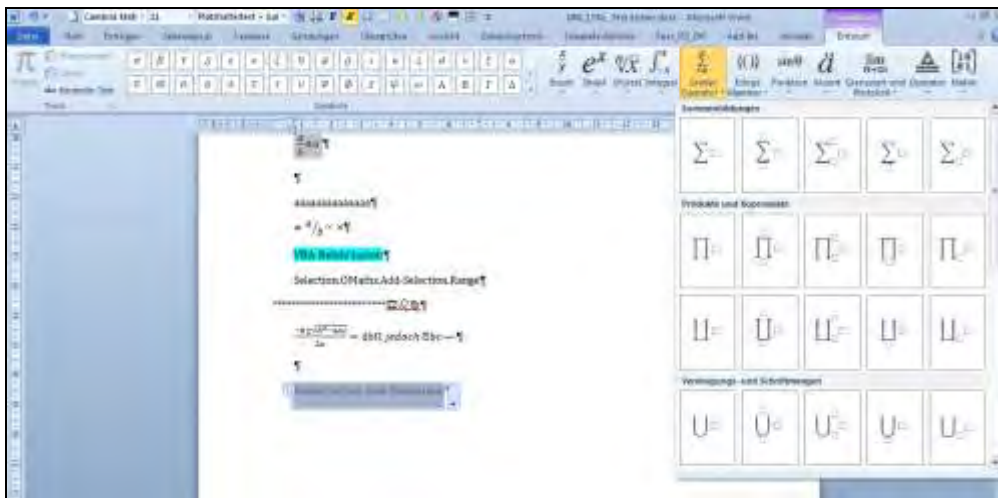


Bild 11.6: Screenshot, Formeleditor, Gruppe "Strukturen"

Bei den "Strukturen" findet man aber auch die "Akzente", mit der man die Symbole modifiziert.

* * *

Achtung: Bei Dateien im Kompatibilitätsmodus ist der rechte Teil mit den Formeln (Gruppe "Strukturen") inaktiv, siehe folgendes Bild (mit einer Ausschnittvergrößerung):

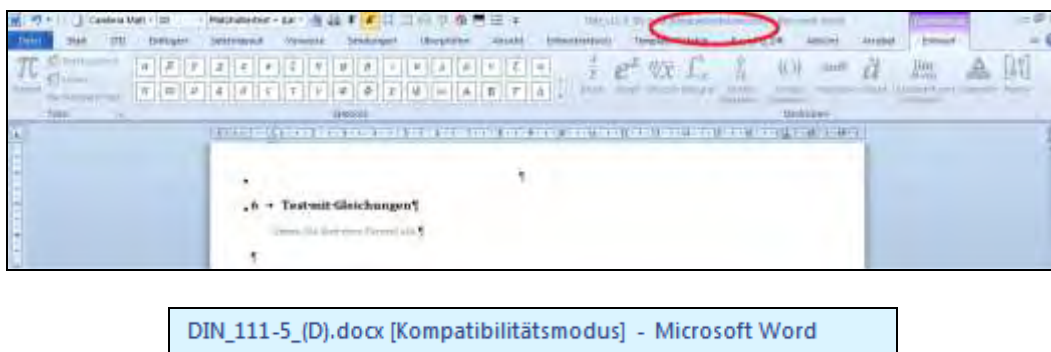


Bild 11.7: Screenshot aus einer Datei im "Kompatibilitätsmodus"

11.2.4 Gruppe "Tools" und das Problem mit dem Buchstaben "a" in Formeln im Gegensatz zu Technischen Zeichnungen mit "a"

Die Gruppe "Tools" befindet sich ganz links im Ribbon-Tab "Formeltools". Dort kann man zwischen "Linear" und "Professionell" wählen. Mit "Professionell" erzeugt man aber eine bessere Lesbarkeit.

Man kann unter "Tools" mit dem Häkchen rechts unten "Formeloptionen" wählen, etwa die Schriftart. Mit Cambria Math hat man ausreichend Auswahl an Formelzeichen und Sonderzeichen.

Aber auch wenn man als Option "Cambria Math" eingestellt hat, sieht im Formeleditor das kursive "a" wie in den meisten Serifen-Schriftarten (etwa Times New Roman und Cambria) anders aus als das nicht-kursive "a", im Gegensatz zu den serifenlosen Schriftarten wie "Arial". In

technischen Zeichnungen müssen die Buchstaben der Normschrift nach den Normen der Reihe DIN EN ISO 3098 entsprechen, wo der kleine Buchstabe **a** eher wie das kursive "a" in der Schriftart "Arial" aussieht.

Abhilfe:

Am besten ist vermutlich, man vermeidet "a" als Formelzeichen ☺. Ansonsten: Textbereich (Buchstabe "a") markieren und oben im Ribbon-Tab (Menüband-Registerkarte) "Formeltools" ganz links unter "Tools" auf "Normaler Text" klicken. Dann kann man die gewünschte Schriftart wählen, siehe Screenshot.

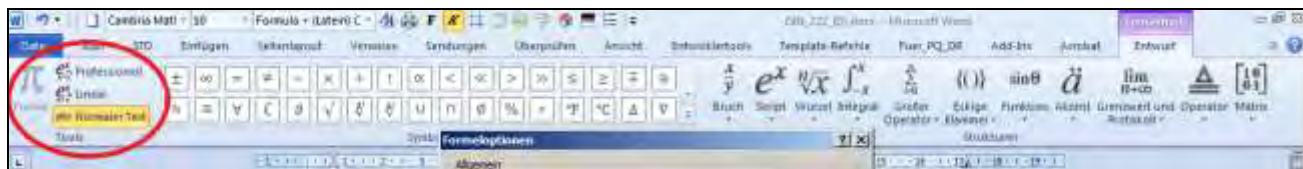


Bild 11.8: Screenshot für "Tools" → "Normaler Text"

Als Schriftart für das "a" empfehle ich hier "Cambria Math"⁹. In **Cambria Math** sieht das kursive "a" wie in "Arial" aus, in **Cambria** selbst sieht es dagegen wie in Times New Roman "a" aus (außer in der Formel, wenn man nicht obigen Trick verwendet).

11.2.5 Probleme beim Speichern der Datei im Fall von "Kompatibilitätsmodus"

Achtung: Aus den Formeln werden Grafiken, wenn man "kompatibel" speichert oder die Datei als ".doc" speichert oder die Formel über die Zwischenablage in eine ".doc"-Datei einfügt. Bei ".docx" im Kompatibilitätsmodus erfolgt die Umwandlung, sobald man sie mit den alten Versionen Word 2000 oder Word 2003 speichert.

$$sndddd \pm abc \cdot aaa + \frac{c^2 \pm \sqrt{aa^2 - 4ac}}{2b}$$

Bild 11.9: Screenshot einer Gleichung, die zur Grafik wurde (vergrößert)

Hingegen bleiben Formeln mit dem alten Formeleditor 3 unverändert, und können weiterhin weiterbearbeitet werden.

⁹ In Word 2000 muss man bei Zeichen mit "Cambria Math" im normalen Text den Zeilenabstand auf "genau" stellen, sonst wird der Zeilenabstand extrem groß.

11.2.6 Unterschied, ob Formel allein steht ("Display Style") oder in einer Zeile mit Text ("Inline Style")¹⁰

Steht eine Formel bei Word 2010 allein in einem Absatz, wird von Word 2010 der "Display-Style" ("abgesetzte Formel") verwendet. Bei Summenzeichen, Integralen usw. stehen die Grenzwerte darunter oder darüber (*hier alles als Screenshots eingefügt*).

$$\int_{i=1}^{\infty} x dt + \sum_{i=1}^n a_i + \prod_{i=1}^n a_i$$

Ansonsten verwendet Word 2010 bei Text (oder Leerzeichen oder Tabulator) in der Zeile automatisch den "Inline-Style". Die Bereichsangabe des Summenzeichens oder eines Integrals usw. steht dann daneben statt darunter und darüber ("Inline-Style").

$$\int_{i=1}^{\infty} x dt + \sum_{i=1}^n a_i + \prod_{i=1}^n a_i \quad (1)$$

Dies kann nicht verhindert werden.

Bei Brüchen ohne Text in Zeile ("Display-Style") ist die Zeichengröße normal:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Mit Text in Zeile ("Inline Style") ist die Größe kleiner, aber immerhin ist das Aussehen dasselbe:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} -b \pm \sqrt{b^2 - 4ac} \quad (2)$$

Nur bei einfachen Formeln und Gleichungen gibt es keinen nennenswerten Unterschied, "Display Style":

$$\vartheta_0 = (26^{+3}_0) \text{ } ^\circ\text{C}$$

"Inline-Style":

$$\vartheta_0 = (26^{+3}_0) \text{ } ^\circ\text{C} \quad (3)$$

Abhilfe:

Am besten Einfügen der Formel in eine **unsichtbare Tabelle** stellen, (*hier ist zur Erläuterung das rechte Tabellenfeld eingefärbt*)¹¹:

$$\sum_{i=1}^n m + x_{-2}^{+3} \quad (4)$$

Keine Lösung fand ich zu folgendem Problem beim Summenzeichen (Sigma) in einem Bruch:

Statt $\frac{\sum_{i=1}^n a}{b} + \sum_{i=1}^n x$ wie mit dem alten Formeleditor 3.0 erhält man jetzt $\frac{\sum_{i=1}^n a}{b} + \sum_{i=1}^n x$.

¹⁰ Auf Deutsch eigentlich "Abgesetzte Formel" und "Formel im Fließtext", aber das sagt keiner.

¹¹ Andere Lösungen: Im Prinzip kann man auch unsichtbare Absatzmarken oder unsichtbare Zeilenumbrüche verwenden. Dies soll aber in manchen Word-Versionen Probleme machen, wurde mir gesagt. Ferner kann man die Gleichung auch in ein Textfeld einfügen.

11.2.7 Speichern einer gut gelungenen Formel mit der AutoKorrektur

Gesamte Formel markieren und oben links anklicken, dann rechten Mausklick, siehe Bild:

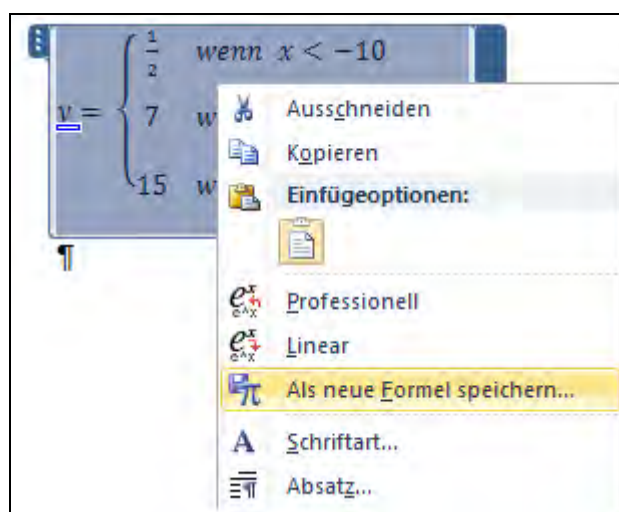
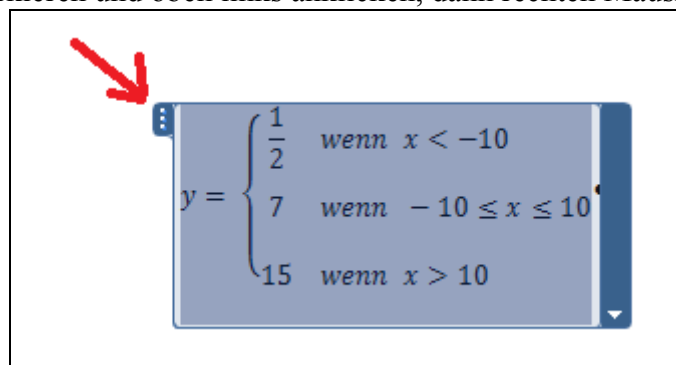


Bild 11.10: Gesamte Formel markieren und rechten Mausklick

Passenden Namen vergeben (hier im Screenshot als Name: "TFormel" und sich diesen merken:

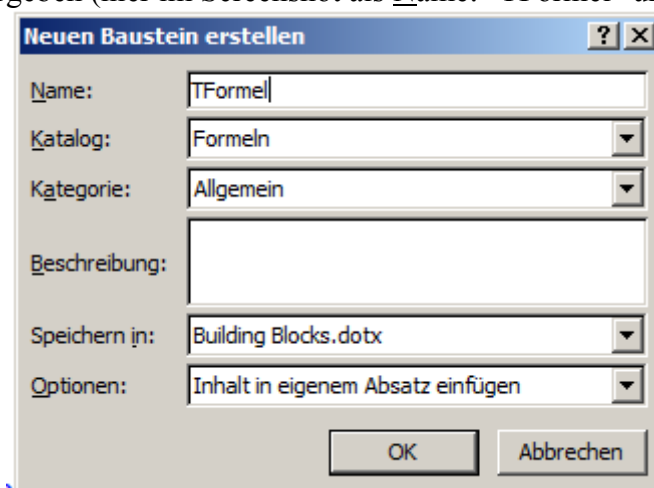



Bild 11.11: Gesamte Formel markieren und rechten Mausklick


Die Formel ist jetzt in den AutoKorrekturen gespeichert (im "Building Blocks.dotx", nicht in der Normal.dotm).

11.3 Anwendungsbeispiel für den Formeleditor in Word 2010: Grenzabmaße übereinander anordnen

Bei Grenzabmaßen, die übereinander stehen sollen, eignet sich in Word 2010 folgendes:
Zuerst leere Formel erzeugen mit "Einfügen → Formel", dann:

Strukturen → Skript →  → $a_{-0,4}^{+0,3}$ [Linear-Ansicht $a_{(-0,4)}^{(+0,3)}$]

Werden in einer Tabellenzelle nur die Grenzabmaße angeben, empfiehlt sich dagegen:

Strukturen → Matrix →  $\begin{matrix} +0,4 \\ -0,5 \end{matrix}$ oder $\begin{matrix} +0,4 \\ 0 \end{matrix}$. [Linear-Ansicht: $\begin{matrix} +0,4 \\ 0 \end{matrix}$]

Man beachte dabei, dass hier die Kommas genau übereinander sind und dass keine gestrichelten Kästchen leer bleiben! Ferner sollten die hoch- und tiefgestellten Zeichen um 2 Punkt kleiner sein als der normale Text.

Komplizierter, aber exakter Lösungsweg für die Spaltenpositionierung von Grenzabweichung, teils mit Nachkommastelle, ohne Leerzeichen:¹²

Einfache, leere Formel erzeugen,
dann Folgendes einfügen (am besten *vorher schreiben und dann reinkopieren*):

$\backslash eqarray(0&,3@-4&)$

(mit nur **einem** Leerzeichen hinter der letzten Klammer",
aber **keinerlei** Leerzeichen davor oder dazwischen!)

"Enter-Taste" drücken oder Umschalten auf "Professionell" ergibt:

0,3
-4

Erklärung:

Das "&" erzeugt Positionierung
(auf dem Bildschirm als dünner senkrechter Strich zu sehen, aber nicht im Druck),
das "@" erzeugt den Zeilenumbruch innerhalb der Formel.
Das **Leerzeichen** am Schluss beendet den Code.
Das **"Enter"** am Schluss startet die Konvertierung.

0,3
-4

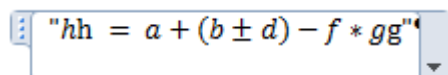
Fertig!

¹² (nach [8], Seite 15, abgewandelt, 6. Nov. 2015).

11.4 Beispiel, in dem der Formeleditor einige Zeichen in Sonderzeichen einer höheren Unicode-Ebene umwandelte

Der neue Formeleditor wandelt die kursiven Buchstaben in buchstabenähnliche Sonderzeichen um, ein kursives "h" wird (fälschlich) durch das Sonderzeichen für "Planck-Konstante" ersetzt.

Im folgenden (unsinnigen) Beispiel wurden die folgenden einfache kursive und einfache nicht-kursive Buchstaben " $hh = a + (b \pm d) - f * gg$ " in eine Formel (Gleichung) in Word 2010 eingefügt:



In der folgenden Prüfung habe ich die Veränderungen blau markiert.

Prüfung mit dem Makro für Zeichenerkennung [3]

Der Unicode von **ħ** ist hexadezimal: 210E und dezimal: 8462,
der genormte englische Zeichenname ist: **PLANCK CONSTANT**.
Der Unicode von **h** ist hexadezimal: 0068 und dezimal: 0104.
Der Unicode von **=** ist hexadezimal: 003D und dezimal: 061.
Der Unicode des Zeichens höherer Ebene **±** ist hexadezimal: 1D44E und dezimal: 119886,
der genormte englische Zeichenname ist: **MATHEMATICAL ITALIC SMALL A**.
Der Unicode von **+** ist hexadezimal: 002B und dezimal: 043.
Der Unicode von **(** ist hexadezimal: 0028 und dezimal: 040.
Der Unicode des Zeichens höherer Ebene **ħ** ist hexadezimal: 1D44F und dezimal: 119887,
der genormte englische Zeichenname ist: **MATHEMATICAL ITALIC SMALL B**.
Der Unicode von **±** ist hexadezimal: 00B1 und dezimal: 0177.
Der Unicode des Zeichens höherer Ebene **đ** ist hexadezimal: 1D451 und dezimal: 119889,
der genormte englische Zeichenname ist: **MATHEMATICAL ITALIC SMALL D**.
Der Unicode von **)** ist hexadezimal: 0029 und dezimal: 041.
Der Unicode von **-** ist hexadezimal: 2212 und dezimal: 8722,
der genormte englische Zeichenname ist: **MINUS SIGN**.
Der Unicode des Zeichens höherer Ebene **ƒ** ist hexadezimal: 1D453 und dezimal: 119891,
der genormte englische Zeichenname ist: **MATHEMATICAL ITALIC SMALL F**.
Der Unicode von ***** ist hexadezimal: 2217 und dezimal: 8727,
der genormte englische Zeichenname ist: **ASTERISK OPERATOR**.
Der Unicode des Zeichens höherer Ebene **ġ** ist hexadezimal: 1D454 und dezimal: 119892,
der genormte englische Zeichenname ist: **MATHEMATICAL ITALIC SMALL G**.
Der Unicode von **g** ist hexadezimal: 0067 und dezimal: 0103.

Hier werden kursive Buchstaben durch Sonderzeichen ersetzt, darunter das kursive "h" durch das Sonderzeichen für die Planck-Konstante.¹³

Ferner sieht man, dass zwar das Pluszeichen unverändert bleibt, andererseits aber das "Minuszeichen" von der Tastatur ("HYPHEN-MINUS") durch das **typografische Minuszeichen** ersetzt wird, ebenso das normale Sternchen durch den "Asterisk-Operator".

Zum richtigen Ersatz von Symbol- und Wingdingszeichen durch Unicode-Zeichen siehe auch [3].

¹³ Bei mir im Büro gab es dadurch zuerst dann Probleme mit einigen alten Druckern, welche diese Sonderzeichen nicht beherrschten.

12 Word-Dateien mittels Word 2010 in eine Pdf-Datei mit "Lesezeichen" umwandeln

12.1 Problem beim Suchen nach Begriffen in Pdf-Dateien, die mit Adobe-Acrobat erstellt wurden

Beim Erzeugen von Pdf-Dateien mittels **Adobe-Acrobat** werden am Zeilenende Silbentrennstriche durch normale Bindestriche ersetzt. Das hat zur Folge, dass man in diesen Pdf-Dateien (bzw. nach Rückkonvertierung in Word) diese Wörter mit der Suchfunktion nicht findet (Version Adobe Acrobat X, Stand Januar 2017).

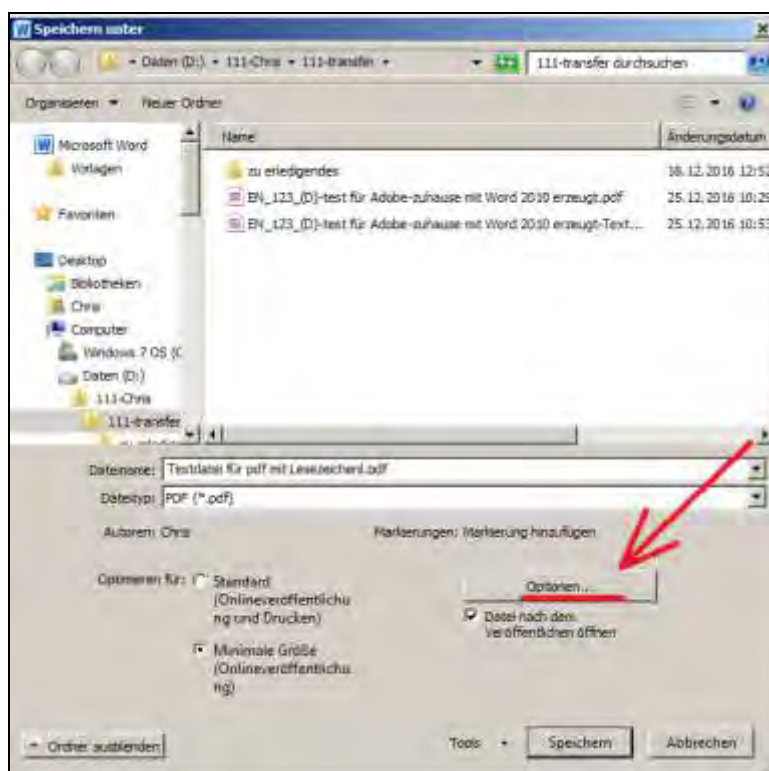
Abhilfe:

Nutzt die **Funktion von Word 2010** (oder höher) zur Erzeugung einer Pdf-Datei, tritt dieser Fehler nicht auf.

12.2 Lesezeichen erzeugen beim Erstellen einer Pdf-Datei mittels Word-Funktion"

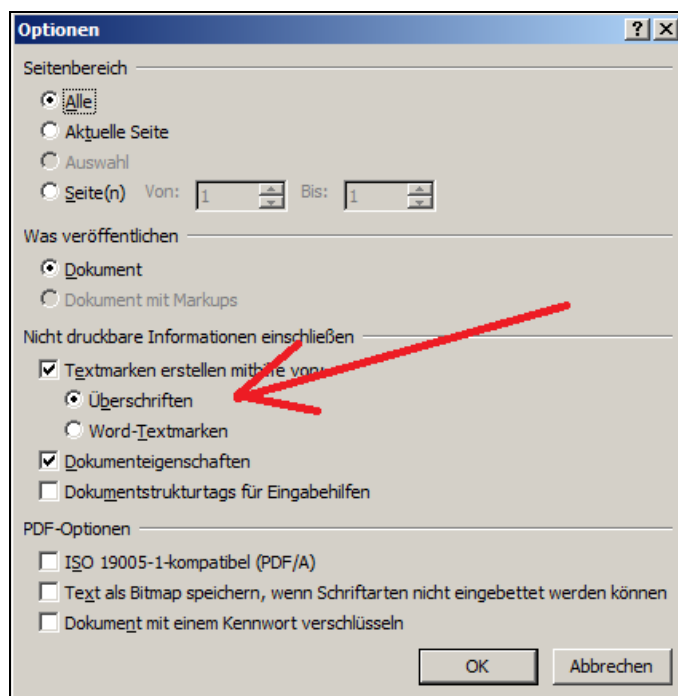
Nutze ich für die Pdf-Erzeugung in Word 2010 die Funktion "Datei → Speichern und Senden → Pdf/XPS-Dokument erstellen...", bekomme ich leider keine "Lesezeichen". Folgende Lösung hierfür fand ich folgende Lösung¹⁴:

Wählen Sie "**Datei → Speichern unter**" und wählen Sie unter *Dateityp* "**PDF**" aus.



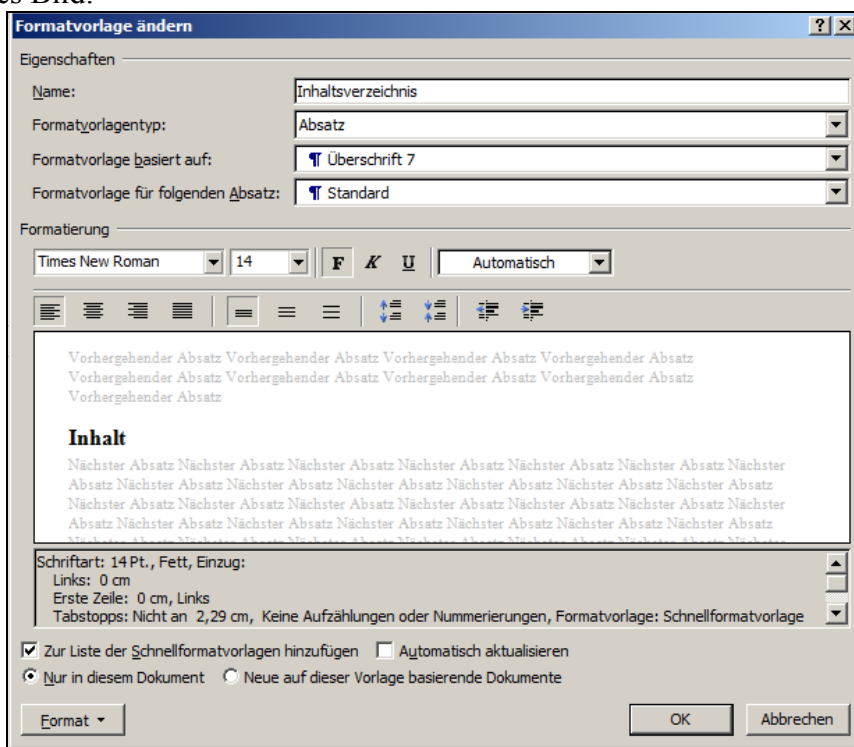
¹⁴ Siehe <http://www.pcs-campus.de/office/word/word-pdf-textmarken>, 2013 | von Katrin Klein.

Klicken Sie dann auf die Schaltfläche *Optionen...* Setzen Sie im sich dann öffnenden Fenster das Häkchen für den Eintrag "Textmarken erstellen mithilfe von:" das Häkchen beim Eintrag "Überschriften".



12.3 Inhaltsverzeichnis bei den Lesezeichen der Pdf-Datei ergänzen

Damit auch das Inhaltsverzeichnis ein Lesezeichen bekommt, habe ich dem Wort "Inhalt" über dem Inhaltsverzeichnis eine Formatvorlage zugeordnet. Diese basiert ("*Formatvorlage basiert auf:*") auf einer Überschriftsebene (hier "7"), welche im Inhaltsverzeichnis nicht vorkommt, siehe folgendes Bild.



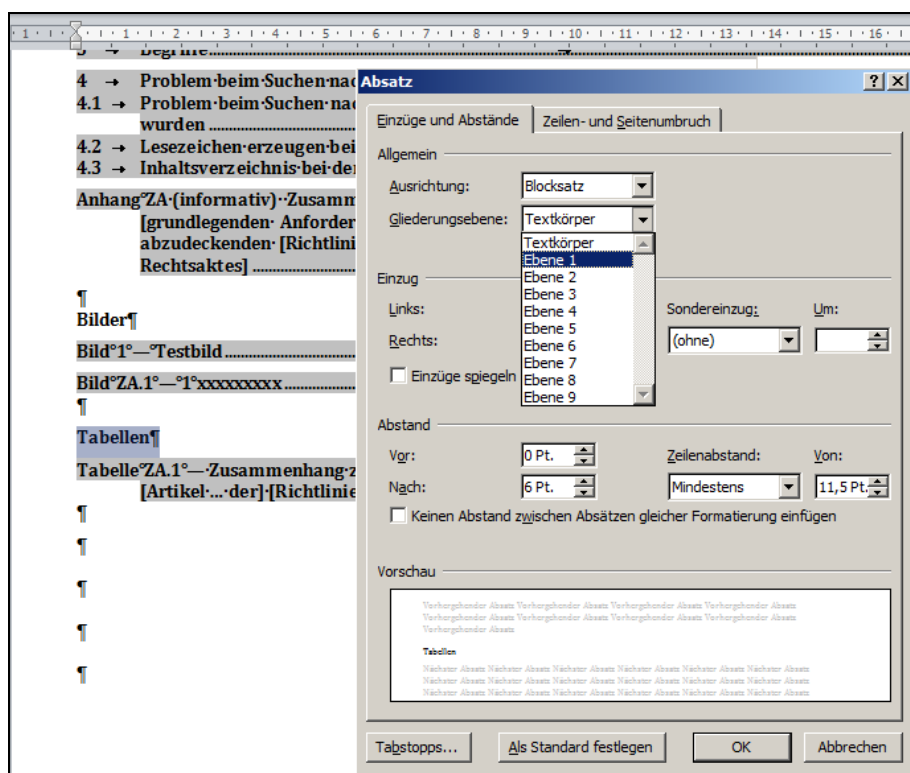
Trotzdem wird sie in den Lesezeichen der künftigen Pdf-Datei vorkommen.

Ferner muss ich die Nummerierung wieder rausnehmen, die Schriftgröße zurückstellen und den Absatz-Abstand zurückstellen. Im Inhaltsverzeichnis taucht das Wort "Inhalt" nicht auf, aber in den "Lesezeichen" der künftigen Pdf-Datei.

Das Entsprechende machte ich bereits bei Anhängen, Literaturhinweisen und dem Impressum, Dort wähle ich bei "Formatvorlage basiert auf..." die "Überschrift 1", da diese im Inhaltsverzeichnis enthalten sein sollen.

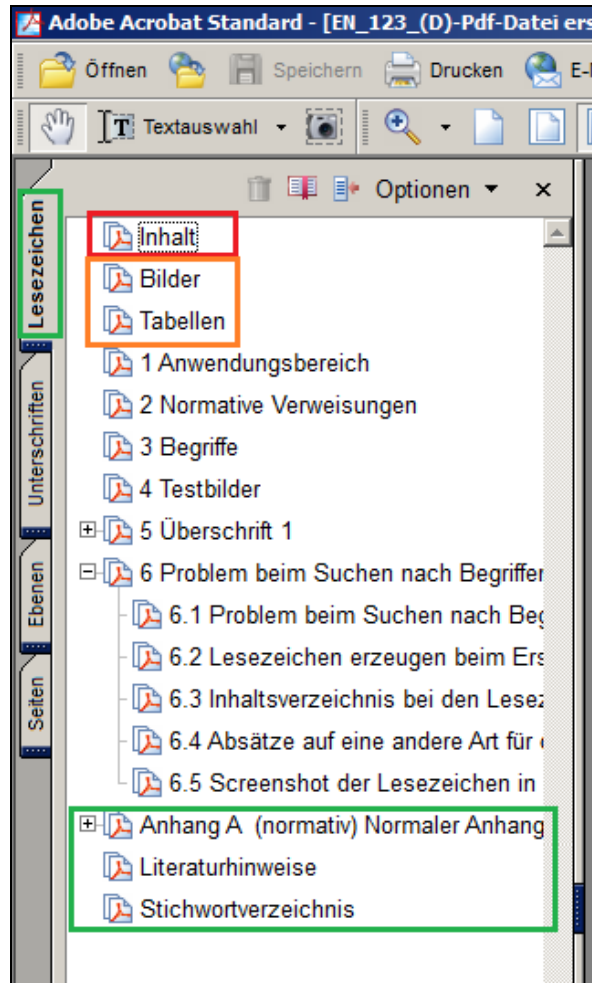
12.4 Absätze auf eine andere Art für die "Lesezeichen" präparieren

Auch die Überschriften des Bilder-Verzeichnisses und des Tabellen-Verzeichnisses kann man für die "Lesezeichen" in Pdf präparieren. Ich wählte hier testweise einen ganz anderen Weg: Rechtsklick auf das zu erfassende Wort "Bilder" und/oder "Tabellen", "Absatz..." wählen, und dann in der Gliederungsebene "Textkörper" durch "Ebene 1" ersetzen, siehe Bild.



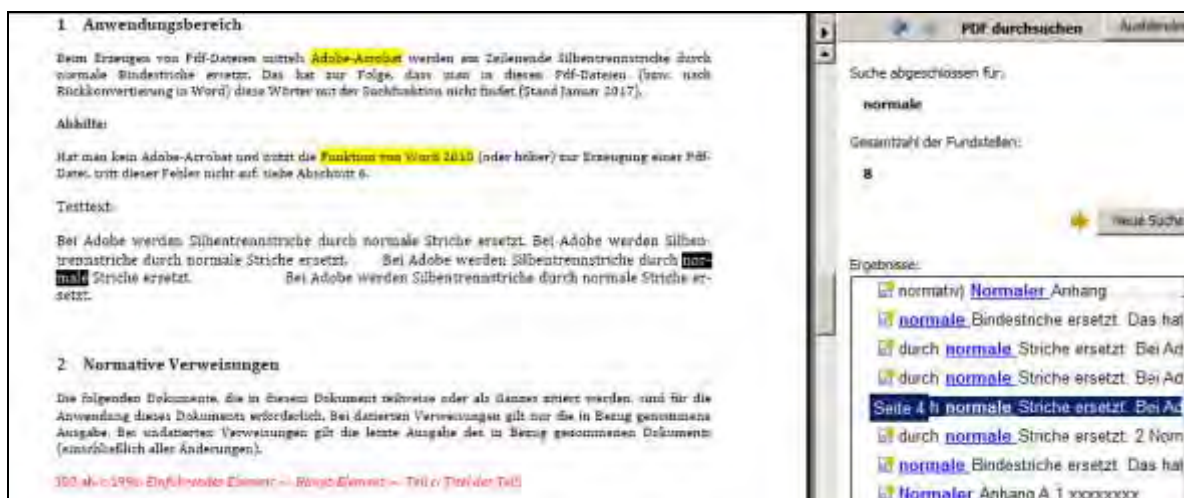
Hier sollte man besser stattdessen eine neue Formatvorlage einführen mit einer nicht-verwendeten "Überschriftsebene" wie bei dem Wort "Inhalt".

12.5 Screenshot der Lesezeichen in der Pdf-Datei



Es sind in den "Lesezeichen" alle Elemente des Inhaltsverzeichnisses vorhanden (grün), ferner das geforderte Element "Inhalt" (rot markiert), siehe 12.3, zusätzlich auch die Elemente "Bilder" und "Tabellen"(orange markiert), siehe 12.4.

Ferner werden bei der Suche in der Pdf-Datei auch Wörter gefunden, die durch Silbentrennung unterbrochen sind (der *Testtext ist nicht in diesem Aufsatz enthalten*):



Damit ist die Pdf-Datei, die direkt aus Word 2010 erzeugt wurde ohne Einsatz von Adobe, besser.

13 Bildschirmansicht invertieren mit "Weiß auf Schwarz"

Will man in Word die Texte auf dem Bildschirm lieber in Weiß auf Schwarz statt in Schwarz auf Weiß angezeigt haben, lässt sich das in den Word-Menüs einstellen.¹⁵ Der Menü-Pfad ist aber völlig anders, als man vermuten würde.

13.1 Weiß auf Schwarz in Word 2010¹⁶

Im Menüband "Seitenlayout", Gruppe "Seitenhintergrund", auf den kleinen Pfeil (rechts) der Schaltfläche "Seitenfarbe" klicken und die "Designfarbe" Schwarz auswählen (es geht auch gut mit Dunkelblau). Der geschriebene Text erscheint jetzt Weiß auf schwarzem Hintergrund, siehe folgenden Screenshot.

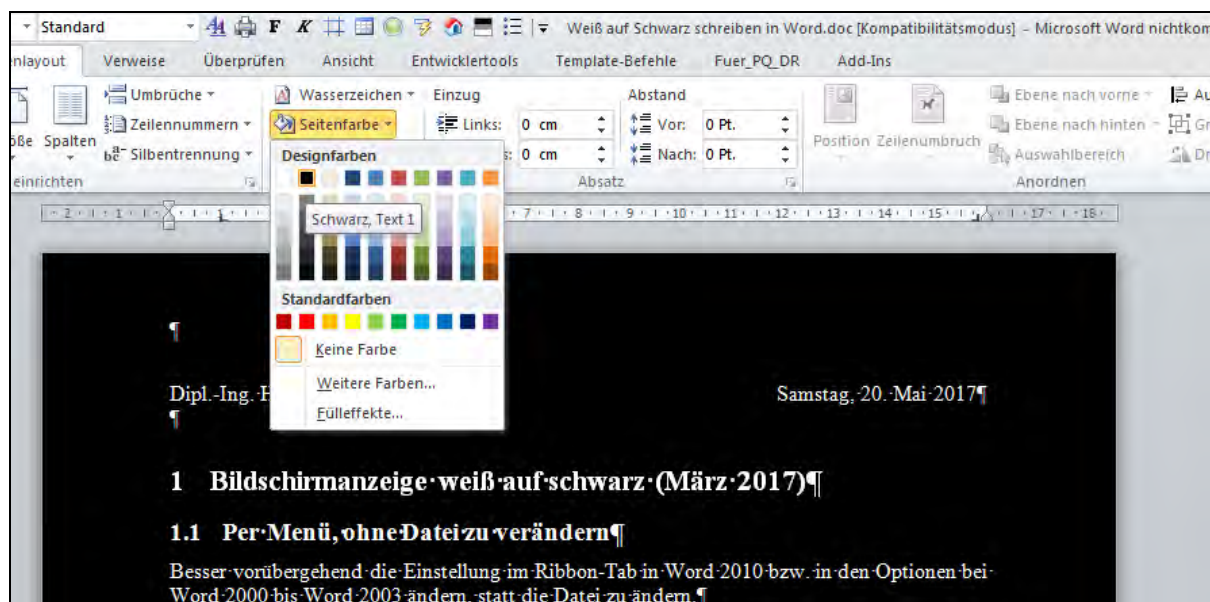


Bild 13.1: Weiß auf Schwarz in Word 2010

13.2 Weiß auf Blau in Word für ältere Apple-Computer

In einer älteren Version von Word für Apple fand ich folgenden Pfad:

- Links oben (links von "Datei") auf "Word" klicken,
- "Einstellungen" anklicken,
- "Allgemein" wählen,
- Haken in "Blauer Hintergrund" setzen.

Siehe auch die folgenden Screenshots für Word 2000.

¹⁵ Für die Augen ist allerdings in der Regel "schwarz auf weiß" günstiger, siehe [6].

¹⁶ <http://www.schieb.de/711424/word-weiss-auf-schwarz-schreiben>, Abruf 18. Mai 2017, modifiziert und Bild ergänzt.

13.3 Weiß auf Blau in Word 2000 bis Word 2003

Bis Word 2003 gibt es die Einstellung "weißer Text auf blauem Hintergrund".

Im Menü "Extras → Optionen" die Registerkarte "Allgemein" wählen und "Blauer Hintergrund, weißer Text" aktivieren, siehe das folgende Bild.

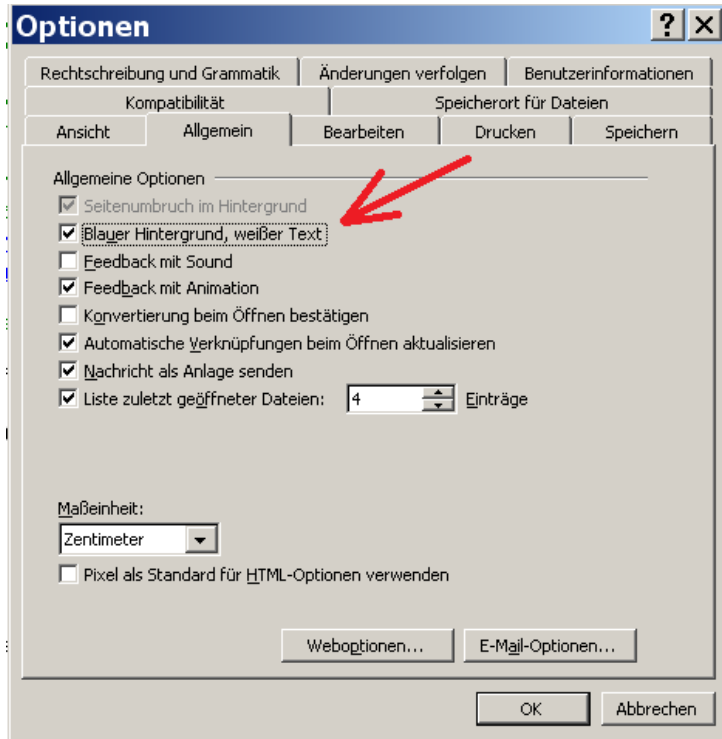


Bild 13.2: Bild: Screenshot von den Optionen in Word 2000 (20. Mai 2017)

Das ergibt dann folgenden Anblick:

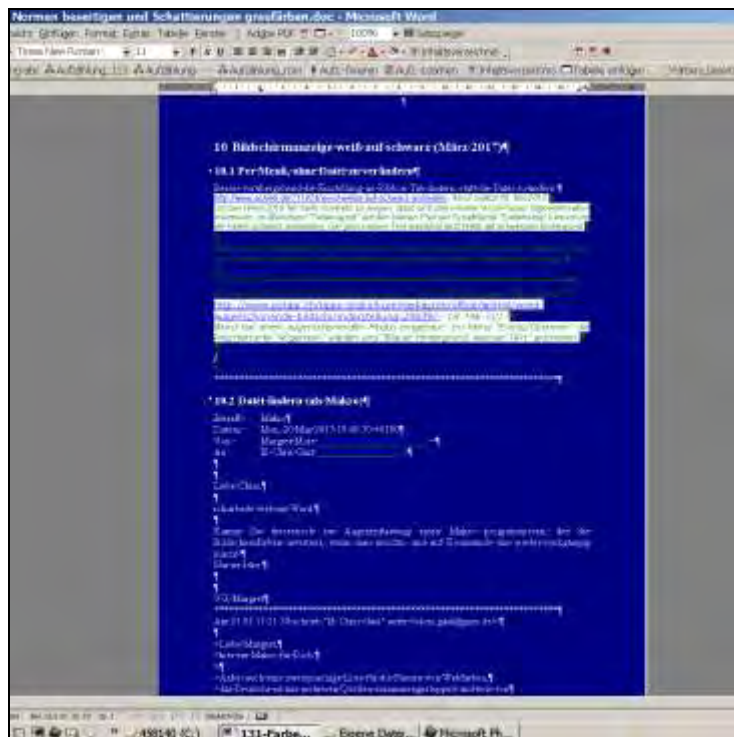


Bild 13.3: Screenshot eines blauen Bildschirms mit weißem Text in Word 2000

14 Zwei Word-Dokumente auf dem Bildschirm vergleichen

Man kann in Word bei einem großen, breiten Bildschirm leicht zwei Word-Dateien nebeneinander anordnen, zum Beispiel zwei Versionen einer Datei oder Original und Übersetzung einer Datei.

Schritt 1: Beide Word-Dateien öffnen.

Schritt 2: Mit Windows-Taste¹⁷ + Pfeil nach links das eine Dokument links anordnen;

Schritt 3: Mit Windows-Taste + Pfeil nach rechts das zweite Dokument rechts anordnen.

15 Bildschirmtastatur

Wer keine griechische oder russische Tastatur besitzt, aber gelegentlich mit den Sprachen am Computer arbeiten will, braucht sich keine spezielle Tastatur dafür zu kaufen, sondern kann die sogenannte "Bildschirmtastatur" benutzen.

15.1 Aufruf der Windows-Bildschirmtastatur unter Windows XP

Für die Bildschirmtastatur ist in Windows XP einzugeben:

Start → Ausführen → Eintippen: "osk" für die Bildschirmtastatur

oder: **Windowstaste** + R → Eintippen: "osk" für die Bildschirmtastatur

oder **Start** → Programme → Zubehör → Eingabehilfen → Bildschirmtastatur"

Oder per Makro, siehe unten.

Achtung: Man muss eventuell beim ersten Mal auch noch die Tastenanzahl einstellen, sonst hat man einige Zeichen wie das Kleiner- und Größer-Zeichen doppelt.

15.2 Bildschirmtastatur unter Windows 7

Klicken Sie auf "**Start**" (bzw. Windows-Logo),

dann auf "**Alle Programme**" klicken,

dann auf "**Zubehör**",

dann auf "**Erleichterte Bedienung**"

und zuletzt auf "**Bildschirmtastatur**" klicken.

Man kann dabei mit einem Rechtsklick auf der Maus ("*Senden an*") eine Verknüpfung auf den Desktop anlegen.

(Stand Juli 2019)


15.3 Bildschirmtastatur unter Windows 10

a) **Windows-Taste** + **R**, dann "osk" eingeben und **Ok** drücken.

b) Ganz links unten auf dem Bildschirm in der Menüleiste drücken ,

"Erleichterte Bedienung" wählen;

dann auswählen "Bildschirmtastatur".

¹⁷ Das ist auf der Tastatur links unten die Taste  zwischen "Strg" und "Alt".

15.5 Makro für den Aufruf der Bildschirmtastatur

Bildschirmtastatur mit einem Makro für Word (VBA) aufrufen:

15.5.1 Makro für Bildschirmtastatur in Windows XP und Windows 7

Private Sub Bildschirmtastatur_für_Windows_7()

```
' aus: http://www.office-loesung.de/ftopic390970\_0\_0\_asc.php  
' (Verfasst: 11. Jun 2010, Download 3. Juni 2013, für Windows XP und Windows 7)  
  Dim sh As Object  
  Set sh = CreateObject("WScript.Shell")  
  sh.Run Environ("windir") & "\\System32\\osk.exe"  
  Set sh = Nothing  
End Sub  
'-----
```

15.5.2 Makro für Bildschirmtastatur in Windows 10

Option Explicit

```
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _  
  (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile As String, _  
  ByVal lpParameters As String, _  
  ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
```

```
Private Declare Function Wow64EnableWow64FsRedirection Lib "kernel32.dll" _  
  (ByVal Enable As Boolean) As Boolean
```

Sub Bildschirmtastatur()

' Funktioniert unter Windows 7 und unter Windows 10

*' ******

' Download 1. Juli 2019. "Private Sub RunOsk_on64Bit()".

' aus: <https://stackoverflow.com/questions/49826995/calling-on-screen-keyboard-from-excel-vba>

' und [https://social.msdn.microsoft.com/Forums/vstudio/en-US/e9252bb7-1df0-4826-880f-](https://social.msdn.microsoft.com/Forums/vstudio/en-US/e9252bb7-1df0-4826-880f-ad0a636a18cd/error-when-opening-onscreen-keyboard-in-windows-7-quotcould-not-start-onscreen-keyboardquot?forum=vbgeneral)

' ad0a636a18cd/error-when-opening-onscreen-keyboard-in-windows-7-quotcould-not-start-onscreen-keyboardquot?forum=vbgeneral

' This is a quick and dirty proof of concept, but the same code works ok both in Windows XP x86

' and Windows 7 x86-64. The On Error Resume Next avoids a runtime when calling

' Wow64EnableWow64FsRedirection() in 32 bit systems, but it would be better checking

' Windows' version before trying to call it. This code is for Visual Basic 6, but I don't see any reason

' impeding to use it in other languages.

' Antonio Grijan, Sunday, February 12, 2012 9:25 AM,

*' ******

Const SW_SHOWNORMAL = 1

On Error Resume Next

Wow64EnableWow64FsRedirection **False**

ShellExecute 0, "open", "osk.exe", "", "C:\windows\system32\osk.exe", SW_SHOWNORMAL

Wow64EnableWow64FsRedirection **True**

End Sub

Anhang A

Problem mit der (mir lästigen) Änderungsverfolgung und fehlende Anzeige in der Statusleiste von Word 2010

Problem: Plötzlich tauchen Randstriche und Änderungsmarkierungen auf, die man gar nicht haben will. Dies passierte mir anfangs oft, wenn ich zum Beispiel zentrieren wollte und statt **Strg** + **E** versehentlich **Strg** + **Shift** + **E** drückte.

Abhilfe:

Tastenkürzel sowohl für Einschalten als auch für Ausschalten des Befehls "Änderungen verfolgen" (nicht verwechseln mit dem Tastenkürzel für das Zentrieren!):

Strg + **Shift** + **E**

Ob man gerade eingeschaltet oder ausgeschaltet hat, kann man nur unten am Bildschirm in der Statusleiste unter dem Word-Text sehen, siehe Bild. Die Buchstaben "ÄND" werden in Word 2000 bzw. Word 2003 beim Ausschalten **grau**. Dort kann man auch draufklicken zum Umschalten.

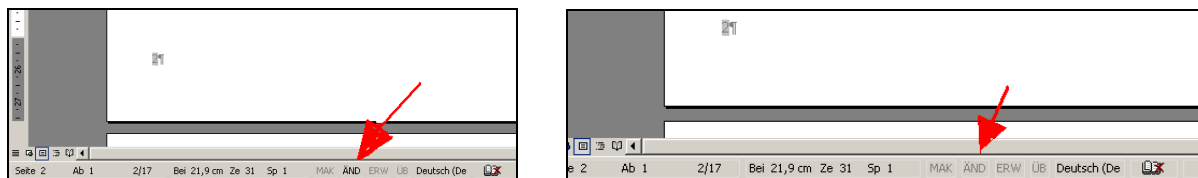


Bild A.1: Screenshot aus Word 2000 bis Word 2003:
Links ist die Änderungsverfolgung eingeschaltet, rechts ausgeschaltet.

Die bereits vorhandene Änderungskennung bleibt bestehen, bis man die Änderungen entweder angenommen oder abgelehnt hat.

Man kann die Änderungskennung auf dem Bildschirm allerdings auch einfach ausblenden und wieder einblenden, ohne sich über Annahme oder Ablehnung zu entscheiden. In Word 2003 geht dies über "Ansicht → Markup" (bzw. mit **Alt** + **A,A**), in Word 2000 über "Extras → Änderungen verfolgen → Änderungen hervorheben" und dann das entsprechende anklicken. In Word 2010 geht dies über das Ribbon "Überprüfen" und dort so ziemlich in der Mitte ("Nachverfolgung"), dort oberste Zeile.

Wenn man das **nicht** beachtet, sieht der Text in der erzeugten Pdf-Datei wie folgt aus:

Man kann die Änderungskennung auf dem Bildschirm allerdings schon vorher ausblenden und wieder einblenden, in Word 2003 über "Ansicht → Markup" (bzw. mit **Alt** + **A**, **A**), in Word 2000 dagegen über "Extras → Änderungen verfolgen → Änderungen hervorheben"

Es kann sein, dass **in Word 2010** die Änderungsverfolgung unten in der Statusleiste nicht angezeigt wird.

Abhilfe:

- Statusleiste mit rechter Maustaste anklicken.
- Dann im Menü "Änderungen nachverfolgen" aktivieren.

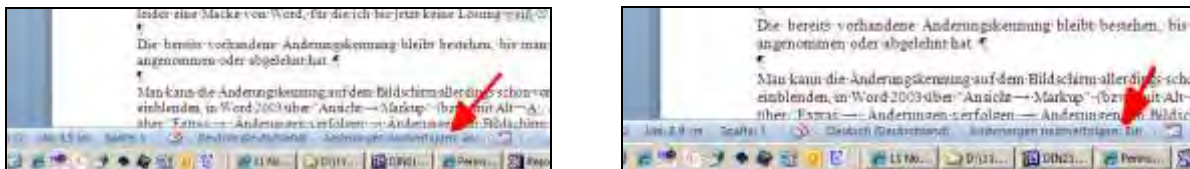


Bild A.2: Screenshot der Taskleiste aus Word 2010:

Links ist im obigen Bild die Änderungsverfolgung eingeschaltet, rechts ist sie ausgeschaltet.

Anwendung

Unten auf dem Bildschirm mit der Maus die Worte anklicken:

"Änderungen nachverfolgen: Aus"

Dann wird dieses gestartet.

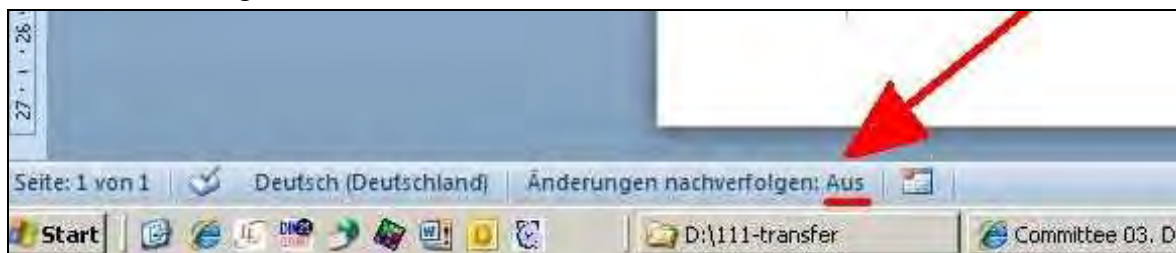


Bild A.3: Änderungsverfolgung starten

Unten auf dem Bildschirm mit der Maus den Text anklicken:

"Änderungen nachverfolgen: Ein"

Dann wird dieses beendet.

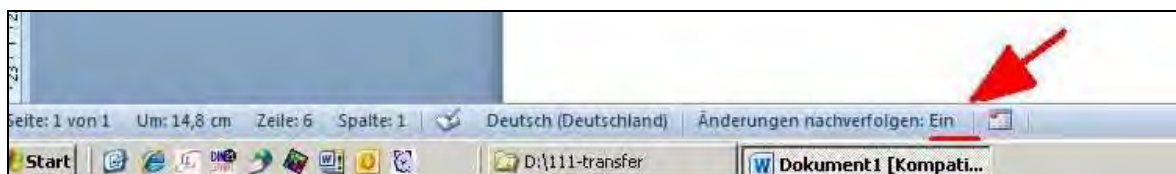


Bild A.4: Änderungsverfolgung beenden

ACHTUNG: Das Icon rechts neben dem Wort "Aus" bzw. "Ein" startet den Makro-Rekorder!

Um Änderungen überhaupt zu sehen, wählt man in Word 2010 oben auf dem Bildschirm die Registerkarte "Überprüfen". Dann wählt man in der Gruppe "Nachverfolgung" eines der Icons aus.

Anhang B

Einige (unnötige) Makros für die AutoKorrektur

Da man so selten die AutoKorrekturen bearbeitet, sind eigentlich Makros dafür nicht sinnvoll. Weil aber der Pfad zu den Einstellungen für die AutoKorrekturen in Word 2010 so umständlich geworden ist, habe ich folgendes Makro entwickelt.

Sub AutoKorrektur_Dialog_aufrufen()

```
' AutoKorrekturen per Dialog einfügen und bearbeiten:  
' die markierte Stelle wird dabei eingelesen als Text, der zukünftig durch die AutoKorrektur erzeugt  
' werden soll.  
' Es wird dabei die Sprachversion der AutoKorrekturen (acl-Datei) ausgewählt, die der markierten  
' Text-Stelle entspricht.
```

```
Dialogs(wdDialogToolsAutoCorrect).Show
```

```
End Sub
```

Man kann sich diesen Befehl auch auf die Schnellstartleiste legen, dann braucht man dafür also kein Makro.

B.1 Makros zum Entfernen von AutoKorrekturen

Sinnvoller sind vielleicht die beiden folgenden Makros für das Entfernen von AutoKorrektur-Einträgen.

Im folgenden Makro muss der "fehlerhafte" Eintrag eingegeben werden wie "**dna**" (für "dann"). Es werden die AutoKorrekturen für Deutsch, Englisch und Französisch durchsucht. Ggf. sind weitere Sprachen (Italienisch) oder Gebiete (Schweiz, Österreich) im Makro zu ergänzen.

Sub einzelnen_AutoKorrektur_Eintrag_löschen()

```
' H. Chris Gast, aus der Hilfe, erheblich modifiziert, 13. November 2012  
' Dieses Makro erfasst deutsche, englische und französische AutoKorrektur-Einträge.
```

```
Dim strEntry As String
```

```
Dim acEntry As AutoCorrectEntry
```

```
Dim intResponse As Integer
```

```
,
```

```
On Error Resume Next
```

```
strEntry = InputBox("Bitte geben Sie den Namen des zu löschenden AutoKorrektur-Eintrages ein.")
```

```
' **** deutsche Einträge:
```

```
Selection.LanguageID = wdGerman
```

```
For Each acEntry In AutoCorrect.Entries
```

```
    If acEntry.Name = strEntry Then
```

```
        intResponse =
```

```
            MsgBox("Wollen Sie den Eintrag wirklich löschen?" & vbCrLf & vbCrLf_
```

```
                & acEntry.Name & " => " & acEntry.Value, vbYesNo)
```

```
                & acEntry.Name & " => " & acEntry.Value, vbYesNo)
```

```
        If intResponse = vbYes Then
```

```
            acEntry.Delete
```

```
        End If
```

```
    Exit Sub
```

```
End If
```

```
Next acEntry
```

```
' ***** us-englische Einträge
```

```
Selection.LanguageID = wdEnglishUS
```

```
For Each acEntry In AutoCorrect.Entries
```

```
    If acEntry.Name = strEntry Then
```

```

intResponse = _
    MsgBox("Wollen Sie den US-Englischen Eintrag wirklich löschen?" & vbCrLf _
        & acEntry.Name & " => " & acEntry.Value, vbYesNo)
If intResponse = vbYes Then
    acEntry.Delete
End If
Exit Sub
End If
Next acEntry
' ***** uk-englische Einträge
Selection.LanguageID = wdEnglishUK
For Each acEntry In AutoCorrect.Entries
    If acEntry.Name = strEntry Then
        intResponse = _
            MsgBox("Wollen Sie den Britisch-Englischen Eintrag wirklich löschen?" & vbCrLf _
                & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
        End If
    Exit Sub
End If
Next acEntry
' ***** französische Einträge
Selection.LanguageID = wdFrench
For Each acEntry In AutoCorrect.Entries
    If acEntry.Name = strEntry Then
        intResponse = _
            MsgBox("Wollen Sie den französischen Eintrag wirklich löschen?" & vbCrLf _
                & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
        End If
    Exit Sub
End If
Next acEntry
' *****
MsgBox "Kein AutoKorrektureintrag gefunden: " & strEntry & vbCrLf _
    & "Versuchen Sie es mit dem Makro ""a3b_einzelnes_AutoKorrektur_Ergebnis_löschen.""
End Sub
' -----

```

Im folgenden Makro wird das ungewünschte AutoKorrektur-Ergebnis eingegeben. Hier wird nach dem fehlerhaften AutoKorrektur-Ergebnis wie "warden" gesucht. Dieses Makro funktioniert allerdings nicht für Sonderzeichen wie Smileys, man kann mit der InputBox nur einfache Zeichen eingeben.

Sub einzelnes_AutoKorrektur_Ergebnis_löschen()

```

' H. Chris Gast, aus der Hilfe, erheblich modifiziert, 6. Juni 2013.
' Dieses Makro erfasst deutsch, englisch und französische AutoKorrektur-Einträge.
' Hier wird jetzt nicht nach dem Eintrag gesucht, sondern nach dem Resultat eines Eintrags.
' Dieses Makro funktioniert allerdings nicht für Sonderzeichen wie Smileys.
Dim strEntry As String
Dim acEntry As AutoCorrectEntry
Dim intResponse As Integer
Dim i As Integer: i = 0
Dim k As Integer: k = 0
'
On Error Resume Next
strEntry = InputBox _
    ("Bitte geben Sie das Ergebnis des zu löschenden AutoKorrektur-Eintrages ein." _

```

```

        & vbCrLf & vbCrLf _
        & " Achtung: Dies funktioniert nur für einfachen Text, nicht für Sonderzeichen wie Smileys.")
' **** deutsche Einträge:
Selection.LanguageID = wdGerman
For Each acEntry In AutoCorrect.Entries
    If acEntry.Value = strEntry Then
        intResponse = _
        MsgBox("Wollen Sie den Eintrag wirklich löschen?" & vbCrLf _
            & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
            i = i + 1
        End If
        k = k + 1
        " Exit Sub ' es kann mehrere Einträge geben!
    End If
Next acEntry
' ***** us-englische Einträge
Selection.LanguageID = wdEnglishUS
For Each acEntry In AutoCorrect.Entries
    If acEntry.Value = strEntry Then
        intResponse = _
        MsgBox("Wollen Sie den US-Englischen Eintrag wirklich löschen?" & vbCrLf _
            & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
            i = i + 1
        End If
        k = k + 1
        " Exit Sub ' es kann mehrere Einträge geben!
    End If
Next acEntry
' ***** UK-englische Einträge
Selection.LanguageID = wdEnglishUK
For Each acEntry In AutoCorrect.Entries
    If acEntry.Value = strEntry Then
        intResponse = _
        MsgBox("Wollen Sie den Britisch-Englischen Eintrag wirklich löschen?" & vbCrLf _
            & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
            i = i + 1
        End If
        k = k + 1
        " Exit Sub ' es kann mehrere Einträge geben!
    End If
Next acEntry
' ***** französische Einträge
Selection.LanguageID = wdFrench
For Each acEntry In AutoCorrect.Entries
    If acEntry.Value = strEntry Then
        intResponse = _
        MsgBox("Wollen Sie den französischen Eintrag wirklich löschen?" & vbCrLf _
            & acEntry.Name & " => " & acEntry.Value, vbYesNo)
        If intResponse = vbYes Then
            acEntry.Delete
            i = i + 1
        End If
        k = k + 1
        " Exit Sub ' es kann mehrere Einträge geben!
    End If
Next acEntry
' *****

```

```

If k = 0 Then
    MsgBox "Keine AutoKorrektureintrag gefunden mit dem Ergebnis: " & strEntry
Elseif i = 0 Then
    " MsgBox "Gefunden, aber nicht gelöscht: " & strEntry
Elseif i = 1 Then
    MsgBox "Es wurde ein Eintrag gelöscht von " & k & " Fundstelle(n)."
Else
    MsgBox "Es wurden " & i & " Einträge gelöscht von " & k & " Fundstelle(n)."
End If
End Sub
'-----

```

B.2 ".acl"-Dateien suchen und erklären

Mit dem folgenden Makro finden Sie im Explorer die "acl"-Dateien, in denen die AutoKorrekturen je nach Sprache gespeichert sind, soweit keine Sonderzeichen betroffen sind.

Das Makro listet in einer (neuen) Datei die gefundenen "acl"-Dateien auf, und gibt die zugehörigen Sprachen an.

Sub acl_Dateien_suchen_und_auflisten_und_erklären()

```

' H. Chris Gast, 13. Nov. 2012, Stand 16. Dezember 2012.
' Word 2000 bis Word 2010. Windows XP und Windows 7.
' Sucht die für die AutoKorrektur nötigen ".acl"-Dateien und gibt ggf. die zugehörige Sprache an.
' zum Beispiel "MSO1031.acl  Deutsch (Deutschland)".
Dim Pfad1 As String
Dim Name1 As String
Dim Rng0 As Range
Dim i As Integer: i = 0
Dim Abfrage As Integer
' *****

On Error Resume Next
Pfad1 = Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
' Erste acl-Datei abrufen.
Name1 = Dir(Pfad1 & Application.PathSeparator & "*.acl")
If Err.Number > 0 Or Name1 = "" Then
    MsgBox "acl-Dateien nicht gefunden!" & vbCrLf _
        & "Versuchen Sie es im Explorer unter:" & vbCrLf _
        & Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
Exit Sub
End If
' *****

' Neues Dokument erstellen:
Application.Documents.Add
Selection.WholeStory
Selection.Collapse Direction:=wdCollapseEnd ' (Briefkopf mit Datum soll ggf. erhalten bleiben)
Selection.TypeParagraph
Selection.TypeParagraph
Set Rng0 = Selection.Range
' *****

' Schleife über alle ".acl"-Dateien mit abgefangenen Fehlern:
Do While Name1 <> "" ' Schleife beginnen.
    If Name1 = "" Then Exit Do
    Selection.TypeText Text:=vbCrLf & Name1 & " "
    On Error Resume Next
' Den Sprachcode aus "Name1" der Markierung zuordnen:
Selection.LanguageID = Val(Mid(Name1, 4))
' Aus der zugeordneten Sprache den Sprachnamen entnehmen,
' aber nur dann, wenn die Zuordnung wirklich funktioniert hat:

```

```

If Selection.LanguageID = Val(Mid(Name1, 4)) Then
    Selection.TypeText Text:=Languages(Selection.LanguageID)
    ' Zum Beispiel ergibt die Zahl "1031": "Deutsch, Deutschland".
Else
    Selection.TypeText Text:="((Sprache ließ sich nicht aufrufen!))"
End If
If Err.Number = 5941 Then
    If Mid(Name1, 4, 4) = 127 Then
        Selection.TypeText Text:=""Mathe-AutoKorrektur-Liste""
    Else
        Selection.TypeText Text:="((Dieser Nummer ist keine Sprache zugeordnet))"
    End If
    On Error GoTo 0
Elseif Err.Number > 0 Then
    Selection.TypeText Text:=vbCrLf & "unklarer Fehler " & Err.Number
    On Error GoTo 0
End If
    i = i + 1
    ' nächsten Namen abfragen:
    Name1 = Dir
Loop
' *****
' Wieder als Sprache "Deutsch" einstellen und Hinweise geben:
Selection.LanguageID = 1031
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeText Text:="Sie können die AutoKorrektur-Dateien (*.acl) " _
    & "auf einen USB-Stick speichern und in einen neuen Computer übertragen." _
    & vbCrLf & vbCrLf _
    & "ACHTUNG: AutoKorrekturen mit Bildern, Sonderzeichen oder Formatierungen werden " _
    & "nicht hier, sondern Sprach-unabhängig in der Normal.dot gespeichert."
' *****
' An den Anfang des Dokuments gehen und Zählung ausgeben:
Rng0.Select
Selection.ParagraphFormat.Alignment = wdAlignParagraphLeft
Selection.TypeText Text:"Es gibt im Ordner " & vbCrLf & """" & Pfad1 & """" & vbCrLf & "insgesamt " _
    & i & " AutoKorrektur-Dateien mit der Endung ""*.acl""." & vbCrLf & vbCrLf
' *****
' Abfrage, ob der Explorer-Ordner mit den ".acl"-Dateien geöffnet werden soll:
Abfrage = MsgBox("Es gibt im Ordner """" & Pfad1 & """" insgesamt " & i & " AutoKorrektur-Dateien." _
    & vbCrLf & vbCrLf _
    & "Wollen Sie diesen Explorer-Ordner öffnen?", vbYesNoCancel + vbDefaultButton2)
' Bei "Ja" kommt "6" heraus, bei "Nein" kommt "7" heraus, bei Abbrechen eine "2".
If Abfrage = 6 Then
    Shell "Explorer.exe /e, " & Pfad1, vbNormalFocus
End If
End Sub
' -----

```

Außer Deutsch, Englisch, Französisch und Italienisch sind (zumindest bei mir) die meisten "acl"-Dateien ohne Inhalt. Sie wurden irgendwann einmal erzeugt, weil die Rechtschreibprüfung von Word die Sprache falsch erkannte.

Achtung: Bei der Übertragung der "acl-Dateien" in einen neuen Computer werden AutoKorrekturen mit Formatierungen (etwa schriftabhängige Sonderzeichen) nicht mit übertragen, da diese in der Normal.dot gespeichert werden!

B.3 Die Makros im Formblatt zur Übertragung von Tastenkürzeln und AutoKorrekturen nach Abschnitt 9.4

Download des Formblattes:

bzw. http://www.siebener-kurier.de/chris-aufsaeetze/Formblatt-Tastenkuerzel+Autokorrekturen_uebertragen.doc
siehe [4].

B.3.1 Screenshot des Formblattes

Für die Übertragung von **Tastenkürzeln** sind die **Buttons** oberhalb der Sternchen-Linie zu verwenden (siehe Bild).



Bild B.1: Das Formblatt

Für die Übertragung von (deutschen) **AutoKorrekturen** sind die **Buttons** unterhalb der Sternchenlinie zu verwenden.

B.3.2 Makros im Modul "ThisDocument"

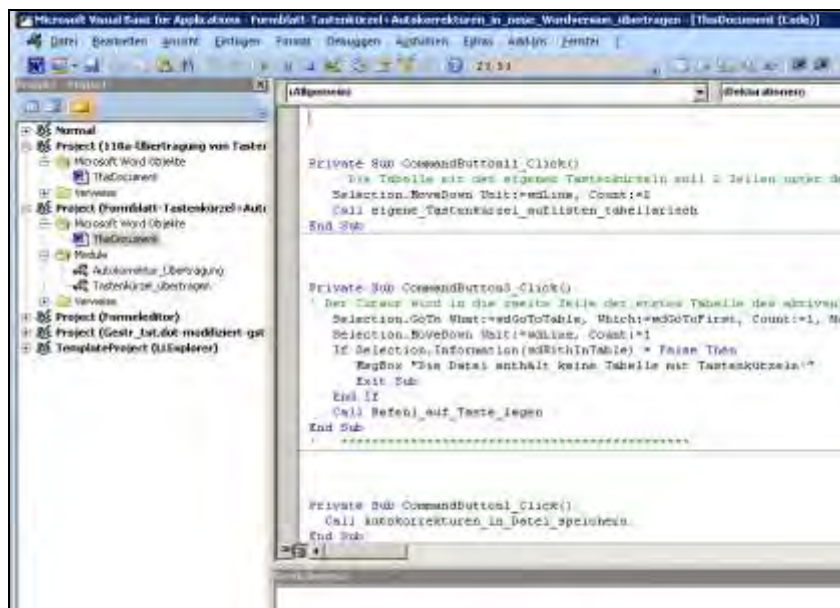


Bild B.2: Screenshot

Private Sub CommandButton11_Click()

' Die Tabelle mit den eigenen Tastenkürzeln soll 2 Zeilen unter dem Button entstehen:

Selection.MoveDown Unit:=wdLine, Count:=2

Call eigene_Tastenkürzel_auflisten_tabellarisch

End Sub

Private Sub CommandButton3_Click()

' Der Cursor wird in die zweite Zeile der ersten Tabelle des aktiven Dokuments gesetzt.

Selection.GoTo What:=wdGoToTable, Which:=wdGoToFirst, Count:=1, Name:=""

Selection.MoveDown Unit:=wdLine, Count:=1

If Selection.Information(wdWithInTable) = False Then

MsgBox "Die Datei enthält keine Tabelle mit Tastenkürzeln!"

Exit Sub

End If

Call Befehl_auf_Taste legen

End Sub

Private Sub CommandButton1_Click()

Call Autokorrekturen_in_Datei speichern

End Sub

Private Sub CommandButton2_Click()

Call Autokorrekturen_aus_Tabelle_in_neuen_Computer_übernehmen

End Sub

Private Sub CommandButton12_Click()

Call acf_Dateien_suchen_und_auflisten_und_erklären

End Sub

B.3.3 Makromodul: **Tastenkürzel übertragen**

Option Explicit

' erzeugt Fehlermeldung, wenn beim Programmieren Dim-Befehle vergessen wurden.
' *****

Sub eigene_Tastenkürzel_auflisten_tabellarisch()

' Dipl.-Ing. Hanna-Chris Gast, Berlin, 21. September 2011, 21. Mai 2013, 20. Januar 2015.
' Unter Benutzung der VBA-Hilfe in Word 2000.
' Die tabellarische Auflistung beginnt an der Stelle des Cursors im Text,
' hier also unter dem Button, der das Makro aufruft.
' Beim Befehl "Speichern unter" sollte die Liste möglichst auf einem USB-Stick gespeichert werden.
' Es wird das Datum angegeben, an dem die Tabelle mit Tastenkürzeln erstellt wurde.

Dim aKey As Variant

'
' *****
' Sicherheitsabfrage:
If ActiveDocument.Tables.Count > 0 **Then**
 MsgBox "Das Dokument enthält bereits eine Tabelle mit Tastenkürzeln!", vbCritical
 Exit Sub
End If
' *****
' Die Tastenkürzel aus der Normal.dot wählen:
CustomizationContext = NormalTemplate
" CustomizationContext = ActiveDocument
,
' Tabelle einrichten, zuerst Tabellenkopf:
ActiveDocument.Tables.Add Range:=Selection.Range, NumRows:=1, _
 NumColumns:=3, DefaultTableBehavior:=wdWord9TableBehavior, _
 AutoFitBehavior:=wdAutoFitFixed
Selection.TypeText Text:="Tastenkürzel"
Selection.MoveRight Unit:=wdCell
Selection.TypeText Text:"Makro- bzw. Befehlsname oder Sonderzeichen"
Selection.MoveRight Unit:=wdCell
 ' Selection.TypeText Text:"Befehlskategorie"
Selection.TypeText Text:"Befehls" & ChrW(31) & "kategorie"
Selection.SelectRow
Selection.Cells.VerticalAlignment = wdCellAlignVerticalCenter
Selection.Font.Size = 11
Selection.Font.Bold = **True**
Selection.Rows.HeadingFormat = **True**
Selection.Rows.Alignment = wdAlignRowLeft
With Selection.ParagraphFormat
 .SpaceBefore = 2
 .SpaceAfter = 2
End With
,
' Die einzelnen Tabellenzeilen mit den Tastenkürzeln und Befehlen:
For Each aKey **In** KeyBindings
 Selection.InsertRowsBelow 1
 Selection.Rows.HeadingFormat = **False**
 Selection.Font.Bold = **False**
 Selection.Font.Size = 10
 With Selection.ParagraphFormat
 .SpaceBefore = 0
 .SpaceAfter = 0
 End With
 Selection.Font.Bold = **False**
 Selection.Collapse
 Selection.InsertAfter aKey.KeyString
 Selection.MoveRight Unit:=wdCell

```

Selection.Font.Bold = False
If aKey.Command = "Symbol" Then
    Selection.InsertAfter Left(aKey.CommandParameter, 1) " aKey.Command & " " &
    Selection.SelectCell
    Selection.Font.Name = Mid(aKey.CommandParameter, 2)
    Selection.Collapse ' Direction:=wdCollapseEnd
Else
    Selection.Font.Name = "Times New Roman"
    Selection.InsertAfter aKey.Command
End If
Selection.MoveRight Unit:=wdCell
Selection.Font.Bold = False
Selection.InsertAfter aKey.KeyCategory
Selection.Collapse Direction:=wdCollapseEnd
Next aKey
'
' Tabelle nachformatieren (für Word 2003 nötig, für Word 2000 unnötig)
Selection.Tables(1).Select
Selection.Font.Bold = False
Selection.Collapse
Selection.SelectRow
Selection.Rows.HeadingFormat = True
Selection.Font.Bold = True
Selection.Collapse
' Buchstabe Eszett statt fälschlich Backslash:
Selection.Tables(1).Columns(1).Select
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "+\"
    .Replacement.Text = "+ß"
    .Forward = True
    .Wrap = wdFindStop
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
' ****
Call Datum_aktualisieren
' ****
' Backup speichern:
With Dialogs(wdDialogFileSaveAs)
    .Name = "Backup_Tastenkürzel_für_Word_" & Format(Date, "yyyy-mm-dd")
    .Show
End With
End Sub
' -----
Private Sub Datum_aktualisieren()
' H. Chris Gast, 25.1.2015, Stand: 26.1.2015.
' Es werden im 3. Absatz von oben Computername und aktuelles Datum eingefügt.
' " ActiveDocument.BuiltInDocumentProperties("Author").Value
ActiveDocument.Paragraphs(3).Range.Select
Selection.TypeText Text:= _
    "Backup aus Computer: " & Environ("COMPUTERNAME") & vbTab & "Speicherdatum: "
Selection.TypeText Text:=Format(Date, "d. MMMM yyyy")
Selection.TypeParagraph

```

End Sub

Sub Befehl_auf_Taste legen()

' H. Chris Gast, www.siebener-kurier.de/chris, Stand 16. September 2011.
' Nach Muster aus dem Internet April 2011 von "Halweg", korrigiert und erheblich erweitert.
' Diese Makro legt einen Befehl, ein Symbol oder ein Makro auf eine Tastenkombination,
' zum Beispiel "AltGr+Ü,ä" (= "Strg+Alt+Shift+Ü, dann ä").
' Der Cursor muss sich bei Aufrufen dieses Makros in der Tabelle mit den Tastenkürzeln und
' Befehlen befinden.

Dim Abfrage, Zeile1

Zeile1:

On Error Resume Next

If Selection.Information(wdWithInTable) = False Then

MsgBox "Markierung ist nicht in der Tabelle!"

Exit Sub

End If

Selection.SelectRow

Abfrage = MsgBox("Wollen Sie diesen Befehl eingeben?", _
vbYesNoCancel + vbQuestion + vbDefaultButton1, "Tastenkürzel")

If Abfrage = vbNo Then

Selection.MoveDown Unit:=wdLine, Count:=1

GoTo Zeile1

End If

If Abfrage = vbCancel Then Exit Sub

Call Einzelzeile

GoTo Zeile1

End Sub

Private Sub Einzelzeile()

' H. Chris Gast, www.siebener-kurier.de/chris, Stand 16. September 2011.
' Einlesen einer einzelnen Tabellenzeile mit Tastenkürzel, Befehlsname und Kategorie:

Dim Tastenkombi(1), Tastenteile, Einzeltaste, Tastenkombination

Dim Befehlsname, Befehlskategorie, Zeichen_Schriftart

Dim Sonderzeichencode As Long

Dim i As Integer, k As Integer

Selection.Collapse

Selection.SelectCell

' Um die Tabellenfeldmarke auszuschließen, muss die Markierung verringert werden.

Selection.MoveLeft Unit:=wdCharacter, Count:=1, Extend:=wdExtend

Tastenkombination = LCase(Selection)

Selection.MoveRight Unit:=wdCell

Selection.SelectCell

Selection.MoveLeft Unit:=wdCharacter, Count:=1, Extend:=wdExtend

Sonderzeichencode = AscW(Selection)

Zeichen_Schriftart = Selection.Range.Font.Name

" Zeichen_Schriftart = Dialogs(wdDialogInsertSymbol).Font

Befehlsname = Selection

Selection.MoveRight Unit:=wdCell

Selection.SelectCell

Selection.MoveLeft Unit:=wdCharacter, Count:=1, Extend:=wdExtend

```

Befehlskategorie = Selection
'
' *****
' Tastenkürzel analysieren:
' Eine Tastenkürzel kann aus zwei Tastenkombinationen ("Tastenkombi") bestehen,
' die hintereinander gedrückt werden, wie "Strg+Alt+ü", dann "Shift ä"
'
' Zuerst Tastenkombinationen zurücksetzen:
Tastenkombi(0) = wdNoKey ' Zahlenwert der ersten Tastenkombination.
Tastenkombi(1) = wdNoKey ' Zahlenwert der ggf. zweiten Tastenkombination.
'
' Kombinationen "Tastenteile" ermitteln, die jeweils gleichzeitig gedrückt werden:
' Vorab vorübergehend umbenennen:
Tastenkombination = Replace(Tastenkombination, "+", "+plus")
Tastenkombination = Replace(Tastenkombination, "num +", "num_plus")
Tastenkombination = Replace(Tastenkombination, "+", "+Komma")
'
Tastenteile = Split(Tastenkombination, ",")
For i = 0 To UBound(Tastenteile)
' Einzeltasten ermitteln:
Einzeltaste = Split(Tastenteile(i), "+") ' die einzelnen zu drückenden Tasten
Tastenkombi(i) = 0
For k = 0 To UBound(Einzeltaste)
Tastenkombi(i) = Tastenkombi(i) + TCode(Einzeltaste(k))
If Tastenkombi(i) >= 10000 Then
MsgBox "Abbruch: Das Tastenkürzel ist unklar."
Selection.SelectRow
Selection.Range.HighlightColorIndex = wdRed
Selection.MoveDown Unit:=wdLine, Count:=1
Exit Sub
End If
Next k
Next i
'
' *****
' Die Tastenkombination dem Befehl zuordnen:
CustomizationContext = NormalTemplate
'***** CustomizationContext = ActiveDocument
On Error GoTo ZeileFehler
'
If Befehlsname = "Symbol" Then GoTo ZeileFehler
If Befehlskategorie <> 6 And Len(Befehlsname) > 1 Then ' (Makro oder sonstiger Befehl)
KeyBindings.Add KeyCategory:=Befehlskategorie, _
KeyCode:=Tastenkombi(0), _
KeyCode2:=Tastenkombi(1), _
Command:=Befehlsname
Selection.SelectRow
Selection.Range.HighlightColorIndex = wdBrightGreen
Else ' (Sonderzeichen)
KeyBindings.Add _
KeyCode:=Tastenkombi(0), _
KeyCode2:=Tastenkombi(1), _
KeyCategory:=wdKeyCategoryCommand, Command:="Symbol", _
CommandParameter:=ChrW(Sonderzeichencode) & Zeichen_Schriftart
Selection.SelectRow
Selection.Range.HighlightColorIndex = wdBrightGreen
End If
Selection.MoveDown Unit:=wdLine, Count:=1
Exit Sub ' (nächste Tabellenzeile umsetzen)
' *****
ZeileFehler:
MsgBox "Tastenkürzel kann nicht zugeordnet werden."
Selection.SelectRow

```

Selection.Range.HighlightColorIndex = wdRed

Selection.MoveDown Unit:=wdLine, Count:=1

Exit Sub

End Sub

Function TCode(Tastenkuerzel)

' H. Chris Gast, www.siebener-kurier.de/chris, Stand 20. Oktober 2011.

' Diese Funktion übersetzt die deutschen Tastennamen in die englischen Word-Key-Codes.

' Nach "Halweg", <http://www.ms-office-forum.net/forum/showthread.php?p=1369186>,

' jedoch völlig verändert und erweitert.

Select Case Tastenkuerzel

' Sondertasten

Case ""

TCode = wdNoKey ' 255

Case "alt"

TCode = wdKeyAlt ' 1024

Case "tab"

TCode = wdKeyTab

Case "return"

TCode = wdKeyReturn

Case "shift"

TCode = wdKeyShift ' 256

Case "umschalt"

TCode = wdKeyShift ' 256

Case "strg"

TCode = wdKeyControl ' 512

Case "pause"

TCode = wdKeyPause

Case "leertaste"

TCode = vbKeySpace

Case "plus"

TCode = wdKeyEquals ' 187 ' eigentlich "+"

Case "#"

TCode = wdKeySlash ' 191

Case "."

TCode = wdKeyPeriod

Case "Komma"

TCode = wdKeyComma ' 188 ' eigentlich ","

Case "end"

TCode = wdKeyEnd ' 35

Case "pos1"

TCode = wdKeyHome

Case "einfügen"

TCode = wdKeyInsert

Case "entf"

TCode = wdKeyDelete ' 46

Case "-"

TCode = wdKeyHyphen ' 189

Case "rechts"

TCode = 39

Case "links"

TCode = 37

Case "bild-ab-taste"

TCode = wdKeyPageDown

Case "bild-auf-taste"

TCode = wdKeyPageUp

Case "abbrechen"

TCode = 3

Case "rück"

```

    TCode = wdKeyBackspace '8
,
Case "akut"
    TCode = wdKeyCloseSquareBrace '221
Case ""
    TCode = 477 ' wdKeyCloseSquareBrace + wdKeyShift
Case "<"
    TCode = 226
Case ">"
    TCode = 226 + wdKeyShift
Case "\"
    TCode = 219 ' wdKeyOpenSquareBrace ' eigentlich "ß"
Case "ß"
    TCode = 219 ' wdKeyOpenSquareBrace
Case "?"
    TCode = 219 + wdKeyShift
Case "^"
    TCode = wdKeyBackSlash
Case "zirkumflex"
    TCode = wdKeyBackSlash
,
' Funktionstasten
Case "f1"
    TCode = wdKeyF1 ' 112
Case "f2"
    TCode = wdKeyF2
Case "f3"
    TCode = wdKeyF3
Case "f4"
    TCode = wdKeyF4
Case "f5"
    TCode = wdKeyF5
Case "f6"
    TCode = wdKeyF6
Case "f7"
    TCode = wdKeyF7
Case "f8"
    TCode = wdKeyF8
Case "f9"
    TCode = wdKeyF9
Case "f10"
    TCode = wdKeyF10
Case "f11"
    TCode = wdKeyF11
Case "f12"
    TCode = wdKeyF12
Case "f13"
    TCode = wdKeyF13
Case "f14"
    TCode = wdKeyF14
Case "f15"
    TCode = wdKeyF15
Case "f16"
    TCode = wdKeyF16 '127
,
' Buchstaben
Case "a"
    TCode = wdKeyA '65
Case "b"
    TCode = wdKeyB
Case "c"
    TCode = wdKeyC
Case "d"

```

```

TCode = wdKeyD
Case "e"
TCode = wdKeyE
Case "f"
TCode = wdKeyF
Case "g"
TCode = wdKeyG
Case "h"
TCode = wdKeyH
Case "i"
TCode = wdKeyI
Case "j"
TCode = wdKeyJ
Case "k"
TCode = wdKeyK
Case "l"
TCode = wdKeyL
Case "m"
TCode = wdKeyM
Case "n"
TCode = wdKeyN
Case "o"
TCode = wdKeyO
Case "p"
TCode = wdKeyP      ' 80
Case "q"
TCode = wdKeyQ
Case "r"
TCode = wdKeyR
Case "s"
TCode = wdKeyS
Case "t"
TCode = wdKeyT
Case "u"
TCode = wdKeyU
Case "v"
TCode = wdKeyV
Case "w"
TCode = wdKeyW
Case "x"
TCode = wdKeyX
Case "y"
TCode = wdKeyY
Case "z"
TCode = wdKeyZ      ' 90
Case "ä"
TCode = wdKeySingleQuote      ' 222
Case "ö"
TCode = wdKeyBackSingleQuote  ' 192
Case "ü"
TCode = wdKeySemiColon        ' 186
Case "ß"
TCode = wdKeyOpenSquareBrace  ' 219
'
' Ziffern
Case "1"
TCode = wdKey1      ' 49
Case "2"
TCode = wdKey2
Case "3"
TCode = wdKey3
Case "4"
TCode = wdKey4      ' 52

```

```

Case "5"
    TCode = wdKey5
Case "6"
    TCode = wdKey6
Case "7"
    TCode = wdKey7
Case "8"
    TCode = wdKey8
Case "9"
    TCode = wdKey9          ' 57
Case "0"
    TCode = wdKey0          ' 48
,
' Sonderzeichen bei den Ziffern in der obersten Reihe:
Case "!"
    TCode = wdKeyShift + wdKey1 ' 256 + 49
Case "````"
    TCode = wdKeyShift + wdKey2 ' 256 + 50
Case "$"
    TCode = wdKeyShift + wdKey3
Case "$"
    TCode = wdKeyShift + wdKey4
Case "%"
    TCode = wdKeyShift + wdKey5
Case "&"
    TCode = wdKeyShift + wdKey6
Case "/"
    TCode = wdKeyShift + wdKey7
Case "("
    TCode = wdKeyShift + wdKey8
Case ")"
    TCode = wdKeyShift + wdKey9 ' 256 + 57
Case "="
    TCode = wdKeyShift + wdKey0 ' 256 + 48
,
' Nummernblock (Ziffernblock)
Case "num 0"
    TCode = wdKeyNumeric0      ' 96
Case "num 1"
    TCode = wdKeyNumeric1      ' 97
Case "num 2"
    TCode = wdKeyNumeric2
Case "num 3"
    TCode = wdKeyNumeric3
Case "num 4"
    TCode = wdKeyNumeric4
Case "num 5"
    TCode = wdKeyNumeric5
Case "num 6"
    TCode = wdKeyNumeric6
Case "num 7"
    TCode = wdKeyNumeric7
Case "num 8"
    TCode = wdKeyNumeric8
Case "num 9"
    TCode = wdKeyNumeric9      ' 105
Case "num *"
    TCode = wdKeyNumericMultiply ' 106
Case "num /"
    TCode = wdKeyNumericDivide  ' 111
Case "num_plus"
    TCode = wdKeyNumericAdd      ' 107 ' eigentlich "num +"
Case "num -"

```

```

TCode = wdKeyNumericSubtract ' 109
Case "num ."
TCode = wdKeyNumericDecimal ' 110
Case Else
TCode = 10000 ' Hoher Zahlenwert, der zur Fehlerabfrage in der Hauptprozedur dient.
End Select
End Function
'-----

```

B.3.4 Makromodul: "Autokorrektur_Übertragung"

```

' Makro-Modul aus der Support.dot, 18. September 2011, korrigiert und verbessert.
' Siehe http://support.microsoft.com/kb/269006/de, dort: "Private Sub btnBackup_Click()".
'
' Hiermit lassen sich Autokorrektur-Einträge zwischen Computern übertragen,
' Funktioniert von Word 2000 bis Word 2010.
Option Explicit
'

```

Sub Autokorrekturen_in_Datei_speichern()

```

' Makro aus der Support.dot entnommen, 18. September 2011, Stand: 28. Januar 2015.
' Creates a new Word document, calls GetAutoCorrectEntries(),
' adds the text "AutoCorrect Backup Document" at the top of the document.
Dim k As Integer
' *****
' Neues Dokument erstellen:
Application.Documents.Add
Selection.WholeStory
Selection.Cut ' löscht ggf. Briefkopf in der Normal.dot.
' *****
' Überschrift für neues Dokument
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeText Text:="AutoCorrect Backup Document"
Selection.TypeParagraph
'add some formatting
ActiveDocument.Paragraphs(4).Range.ParagraphFormat.Alignment = wdAlignParagraphCenter
With ActiveDocument.Paragraphs(4).Range
.Bold = True
.Font.Size = 14
End With
' *****
' Unterprozedur, die Tabelle mit Einträgen erzeugt:
Call GetAutoCorrectEntries
' *****
Selection.HomeKey Unit:=wdStory
' Computernamen und Datum:
' Es werden im 1. Absatz von oben Computernamen und aktuelles Datum eingefügt.
ActiveDocument.Paragraphs(1).Range.Select
Selection.Text = _
"Backup aus Computer: " & Environ("COMPUTERNAME") & vbTab & "Speicherdatum: " _
& Format(Date, "dd.mm.yyyy")
Selection.Collapse Direction:=wdCollapseEnd
Selection.TypeParagraph
' *****
' Vor MsgBox etwa 0,1 Sekunden warten, bis sich Datum zuverlässig generiert hat:
Dim tt As Single
tt = Timer

```

```

Do While Timer < tt + 0.1 'Sekunden
  DoEvents
Loop
' *****
MsgBox "Bitte prüfen Sie diese Tabelle Ihrer Autokorrekturen " _
  & " und speichern Sie diese auf einen USB-Stick oder in einem Netzlaufwerk, " _
  & " von wo Sie im neuen Computer darauf zugreifen können." _
  & vbCrLf & vbCrLf _
  & "Für andere Sprache als Deutsch müssen auch die ""acl-Dateien"" gespeichert werden! " _
  & "Nach dem Speichern öffnet sich dazu ein passendes Unterprogramm."
' *****
' Speicher-Dialog:
With Dialogs(wdDialogFileSaveAs)
  .Name = "AutoCorrect_Backup_Document_" & Format(Date, "yyyy-mm-dd")
  .Show
End With
' *****
' acl-Dateien suchen und mit Erklärung unter der Tabelle auflisten:
Call acl_Dateien_unter_Tabelle_auflisten
End Sub
' -----

```

Private Sub GetAutoCorrectEntries()

```

' aus Support.dot im September 2011, geändert von H. Chris Gast, 25.1.2015 +26.1.2015.
' Inserts each AutoCorrect entry into a Word document including the Value and whether
' or not it is to include formatting. Name is the current name of the entry,
' Value is the text value of the entry, and RTF is a Boolean that is True if the entry
' has "formatted text" checked. Then converts the text into a table and adds a heading
' row to the table.
'
' j is used to loop through the AutoCorrect entries.
' TotalACEntries is the number of AC entries.
Dim j As Long ' Tabellen-Reihen und -Zeilen verlangen "Long"
Dim TotalACEntries As Integer
Dim oTable As Table
Dim IntRow As Integer ' aktuelle Zeile
Dim aRange As Range
Dim BCheckLanguage As Boolean
' *****
' hier muss unbedingt auf die Sprache geachtet werden (ergänzt 14. Mai 2012):
  Selection.LanguageID = wdGerman
' Selection.LanguageID = wdEnglishUK
' Selection.LanguageID = wdEnglishUS
' Selection.LanguageID = wdFrench
' Die Sprache darf sich nicht verstellen!!!
  BCheckLanguage = Application.CheckLanguage
  Application.CheckLanguage = False
' *****
' Tabelle erzeugen:
  TotalACEntries = Application.AutoCorrect.Entries.Count
  Set oTable = ActiveDocument.Tables.Add _
    (Range:=Selection.Range, _
    NumRows:=(TotalACEntries + 1), NumColumns:=3, _
    DefaultTableBehavior:=wdWord9TableBehavior, _
    AutoFitBehavior:=wdAutoFitWindow)
  oTable.Range.Font.Size = 10
  oTable.Range.Rows.Alignment = wdAlignRowLeft
" With oTable.ParagraphFormat
"   .SpaceBefore = 2
"   .SpaceAfter = 2
" End With

```

```

'
' Tabellenkopf einrichten ohne zu markieren:
oTable.Cell(1, 1).Range.Text = "Name"
oTable.Cell(1, 2).Range.Text = "Value"
oTable.Cell(1, 3).Range.Text = "Forma" & ChrW(31) & "tiert"
'
' Die einzelnen Tabellenzeilen mit den AutoKorrekturen:
For j = 1 To TotalACEntries
" oTable.Rows(j + 1).Range.Font.Size = 10
" oTable.Rows(j + 1).Range.Bold = False
" oTable.Rows(j + 1).Alignment = wdAlignRowLeft
' 1. Feld jeder Tabellenzeile:
Set aRange = oTable.Cell(j + 1, 1).Range
aRange.Collapse
aRange.Text = Application.AutoCorrect.Entries.Item(j).Name
" If aRange.Text = "a!Tabellentest" Then
" aRange.Select
" End If
'
' 2. Feld:
Set aRange = oTable.Cell(j + 1, 2).Range
aRange.Collapse
'check for formatting and insert RTF text if necessary
If Application.AutoCorrect.Entries.Item(j).RichText = True Then
Application.AutoCorrect.Entries(j).Apply aRange
Else
aRange.Text = Application.AutoCorrect.Entries.Item(j).Value
End If
'
' 3. Feld:
oTable.Cell(j + 1, 3).Range.Text = _
Application.AutoCorrect.Entries.Item(j).RichText ' (= Wahr oder = Falsch)
' update status bar
Application.StatusBar = "AutoKorrektur " & j & " von" & ChrW(160) _
& TotalACEntries & " wird gerade aufgelistet."
Next j
' *****
' Kopfzeile formatieren:
oTable.Rows(1).Range.Font.Size = 11
oTable.Rows(1).Range.Font.Bold = True
oTable.Rows(1).HeadingFormat = True
With oTable.Rows(1).Range.ParagraphFormat
.SpaceBefore = 3
.SpaceAfter = 3
.Alignment = wdAlignParagraphCenter
End With
' *****
' Spracherkennungs-Einstellung zurückstellen:
Application.CheckLanguage = BCheckLanguage
End Sub
' -----

```

Private Sub acf_Dateien_unter_Tabelle_auflisten()

```

' H. Chris Gast, 13. Nov. 2012, 16. Dezember 2012, 28. Januar 2015.
' Word 2000 bis Word 2010, getestet Windows XP und Windows 7.
' Sucht die für die AutoKorrektur nötigen ".acf"-Dateien und gibt ggf. die zugehörige Sprache an.
' zum Beispiel "MSO1031.acf Deutsch (Deutschland)".

```

```

Dim Pfad1 As String
Dim Name1 As String
Dim Rng0 As Range
Dim i As Integer: i = 0

```

```

Dim Abfrage As Integer
' *****
' Am Dokument-Ende (hinter Tabelle):
Selection.EndKey Unit:=wdStory
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeParagraph
Set Rng0 = Selection.Range
' *****

On Error Resume Next
Pfad1 = Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
' Erste .acl-Datei abrufen.
Name1 = Dir(Pfad1 & Application.PathSeparator & "*.acl")
If Err.Number > 0 Or Name1 = "" Then
MsgBox "acl-Dateien für AutoKorrekturen nicht gefunden!" & vbCrLf _
& "Versuchen Sie es im Explorer unter:" & vbCrLf _
& Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office"), _
vbCritical
Selection.TypeText "acl-Dateien für AutoKorrekturen nicht gefunden!" & vbCrLf _
& "Versuchen Sie es im Explorer unter:" & vbCrLf _
& Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
ActiveDocument.Save
Exit Sub
End If
' *****
' Schleife über alle ".acl"-Dateien mit abgefangenen Fehlern:
Do While Name1 <> "" ' Schleife beginnen.
If Name1 = "" Then Exit Do
Selection.TypeText Text:=vbCrLf & Name1 & " "
On Error Resume Next
' Den Sprachcode aus "Name1" der Markierung zuordnen:
Selection.LanguageID = Val(Mid(Name1, 4))
' Aus der zugeordneten Sprache den Sprachnamen entnehmen,
' aber nur dann, wenn die Zuordnung wirklich funktioniert hat:
If Selection.LanguageID = Val(Mid(Name1, 4)) Then
Selection.TypeText Text:=Languages(Selection.LanguageID)
' Zum Beispiel ergibt die Zahl "1031": "Deutsch, Deutschland".
Else
Selection.TypeText Text:="((Sprache ließ sich nicht aufrufen!))"
End If
If Err.Number = 5941 Then
If Mid(Name1, 4, 4) = 127 Then
Selection.TypeText Text:=""Mathe-AutoKorrektur-Liste""
Else
Selection.TypeText Text:="((Dieser Nummer ist keine Sprache zugeordnet))"
End If
On Error GoTo 0
Elseif Err.Number > 0 Then
Selection.TypeText Text:=vbCrLf & "unklarer Fehler " & Err.Number
On Error GoTo 0
End If
i = i + 1
' nächsten Namen abfragen:
Name1 = Dir
Loop
' *****
' Wieder als Sprache "Deutsch" einstellen und Hinweise geben:
Selection.LanguageID = 1031
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeText Text:="Sie können die AutoKorrektur-Dateien (*.acl) " _
& "auf einen USB-Stick speichern und in einen neuen Computer übertragen." _
& vbCrLf & vbCrLf _

```

```

    & "ACHTUNG: AutoKorrekturen mit Bildern, Sonderzeichen oder Formatierungen werden " _
    & "nicht in den .acl-Dateien, sondern Sprach-unabhängig in der Normal.dot gespeichert, " _
    & "siehe in obiger Tabelle die Einträge mit 'Formatiert' = 'Wahr' (bzw. 'True')."
' *****
' Hinweis auf Formblatt mit Hyperlink:
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeText Text:= _
    "Zum Einlesen der Tabelle in einen neuen Computer wird das Makro-Formblatt benötigt, " _
    & "Download: "
Selection.TypeParagraph
ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, Address:= _
    "http://www.siebener-kurier.de/chris-aufsaeetze/Formblatt-
    Tastenkuerzel+Autokorrekturen_uebertragen.doc", _
    TextToDisplay:= _
    "http://www.siebener-kurier.de/chris-aufsaeetze/Formblatt-
    Tastenkuerzel+Autokorrekturen_uebertragen.doc"
Selection.TypeParagraph
' *****
' An den Anfang der Auflistung gehen und Zählung ausgeben:
Rng0.Select
Selection.ParagraphFormat.Alignment = wdAlignParagraphLeft
Selection.TypeText Text:="Es gibt im Ordner " & vbCrLf & """" & Pfad1 & """" & vbCrLf & "insgesamt " _
    & i & " AutoKorrektur-Dateien mit der Endung ""'.acl''":" & vbCrLf & vbCrLf
' *****
' Abfrage, ob der Explorer-Ordner mit den ".acl"-Dateien geöffnet werden soll:
Abfrage = MsgBox("Es gibt im Ordner """" & Pfad1 & """" insgesamt " & i & ChrW(160) _
    & "AutoKorrektur-Dateien." _
    & vbCrLf & vbCrLf _
    & "Wollen Sie diesen Explorer-Ordner öffnen?", vbYesNoCancel + vbDefaultButton2)
' Bei "Ja" kommt "6" heraus, bei "Nein" kommt "7" heraus, bei Abbrechen eine "2".
If Abfrage = 6 Then
    Shell "Explorer.exe /e, " & Pfad1, vbNormalFocus
End If
End Sub
' -----

' *****
' *****

Sub Autokorrekturen_aus_Tabelle_in_neuen_Computer_übernehmen()
' Makro aus der Support.dot von 2011 entnommen, 18. September 2011.
' Siehe http://support.microsoft.com/kb/269006/de, dort: " Private Sub btnRestore_Click()".
' First warns the user that this will replace their existing entries.
' Then displays the FileOpen dialog box to get the name of an existing AutoCorrect backup
' document. If successful it calls RestoreACEntries().
Dim ACFileName As String
Dim j As Integer
Dim Warnung As Integer
Dim IntAbfrage As Integer
' *****
' Datei mit Autokorrektur-Tabelle öffnen und ihr einen Namen ("ACFileName") zuordnen:
MsgBox "Bitte wählen Sie die Datei aus, in der die gewünschten AutoKorrekturen aufgelistet sind."
With Dialogs(wdDialogFileOpen)
    .Name = "*.doc"
    .Display
    ACFileName = .Name
End With
'
If OpenACDoc(ACFileName) = False Then 'error

```

```

' Restore Entries, call RestoreACEntries() user defined
MsgBox "Fehler! Keine Datei geöffnet!", vbCritical
Exit Sub
End If
'
' Geöffnete Datei mit Tabelle der AutoKorrekturen prüfen:
" If ActiveDocument.Words(1) <> "AutoCorrect "
If ActiveDocument.Tables.Count <> 1 Then
    Warnung = MsgBox("Diese Datei ist kein Original-Autokorrektur-Backup!", _
        vbCritical, "Autokorrektur - Übertragung")
    Exit Sub
End If
If Left(ActiveDocument.Tables(1).Range.Cells(1).Range.Text, 4) <> "Name" _
And Left(ActiveDocument.Tables(1).Range.Cells(2).Range.Text, 5) <> "Value" Then
    Warnung = MsgBox("Diese Datei ist kein Original-Autokorrektur-Backup!", _
        vbCritical, "Autokorrektur - Übertragung")
    Exit Sub
End If
' *****
' Bei großen Listen abfragen, ob vorhandene deutsche AutoKorrekturen gelöscht werden sollen:
If ActiveDocument.Tables(1).Range.Rows.Count > 80 Then
    ' Löschen der deutschen Einträge:
    Dim StrMsgBoxText As String
    Dim strAK As AutoCorrectEntry
    '
    StrMsgBoxText = "Wollen Sie alle vorhandenen deutschen AutoKorrektur-Einträge in diesem " _
        & "Computer löschen?" _
        & vbCrLf & vbCrLf _
        & "Der Inhalt der Datei ""MSO1031.ac1"" und die formatierten AutoKorrekturen " _
        & "in der Normal.dot bzw. Normal.dotm im jetzigen Computer geht dabei unwiderruflich " _
        & "verloren. Dies ist aber zu empfehlen, wenn Sie dafür alle (!) AutoKorrektur-" _
        & "Einträge aus dem Backup hier einfügen."
    IntAbfrage = MsgBox(StrMsgBoxText, vbYesNoCancel + vbDefaultButton2 + vbCritical, _
        "AutoKorrektur-Einträge")
    ' Bei "Ja" kommt "6" heraus, bei "Nein" kommt "7" heraus, bei Abbrechen eine "2".
    If IntAbfrage = 6 Then
        ' Achtung: Es wird die Backupdatei "MSO1031.ac1" geleert, die auch dann leer bleibt,
        ' wenn man später die Abfrage "Änderungen der Normal.dot speichern" mit "Nein"
        ' beantwortet!
        Selection.LanguageID = wdGerman
        For Each strAK In Application.AutoCorrect.Entries
            strAK.Delete
        Next strAK
        ElseIf IntAbfrage = 2 Then
            Exit Sub
        End If
    ' *****
Else
    ' Abfrage, ob gleichnamige AutoKorrektur-Einträge ersetzt werden sollen:
    IntAbfrage = MsgBox("Ihre jetzigen AutoKorrektur-Einträge werden überschreiben, " _
        & "soweit diese denselben Namen haben wie die Einträge in der Tabelle. " _
        & "Sind sie einverstanden?", vbYesNo + vbCritical + vbDefaultButton2, _
        "Autokorrektur - Übertragung")
    If IntAbfrage = vbNo Then Exit Sub
End If
' *****
Call RestoreACEntries
" ActiveDocument.Close SaveChanges:=wdDoNotSaveChanges
End Sub
' -----

```

Private Sub RestoreACEntries()

' Diese Prozedur liest die Tabelle mit den gespeicherten Autokorrekturen des alten Computers aus.

Dim i, NumRows As Integer

Dim oACorrect, Warnung

Dim szName As String

Dim szValue As String

Dim szRTF As String

,

Err.Clear

On Error GoTo RestoreACEntriesErrors:

Set oACorrect = Application.AutoCorrect.Entries

' *****

' hier muss ggf. auf die Sprache geachtet werden (ergänzt 14. Mai 2012):

Selection.LanguageID = wdGerman

' Selection.LanguageID = wdEnglishUK

' Selection.LanguageID = wdEnglishUS

' Selection.LanguageID = wdFrench

' *****

NumRows = ActiveDocument.Tables(1).Rows.Count

Selection.GoTo What:=wdGoToTable, Which:=wdGoToFirst

Selection.MoveRight Unit:=wdCell, Count:=3

,

For i = 2 To NumRows

szName = Selection.Text ' 1. Spalte

Selection.MoveRight Unit:=wdCell ' 2. Spalte

szValue = Selection.Text

Selection.MoveRight Unit:=wdCell ' 3. Spalte

szRTF = Selection.Text

,

If szRTF = "False" Or szRTF = "Falsch" Then

oACorrect.Add Name:=szName, Value:=szValue

Else

Selection.MoveLeft Unit:=wdCell

oACorrect.AddRichText szName, Selection.Range

Selection.MoveRight Unit:=wdCell

End If

Selection.MoveRight Unit:=wdCell

Next i

,

Selection.Rows.Delete

,

MsgBox "Der Vorgang der Übernahme von Autokorrekturen ist abgeschlossen."

Exit Sub

' *****

RestoreACEntriesErrors:

If Err.Number <> 0 Then

MsgBox "Fehler! " & " Die Tabelle ist möglicherweise fehlerhaft. Das Makro wurde abgebrochen." _
& vbCr & Err.Number & " " & Err.Description & " " & szName, vbCritical

End If

End Sub

' -----

Public Function SaveACDoc()

Dim Title As String

,

SaveACDoc = True

Err.Clear

On Error GoTo SaveACDocErrors

,

'ActiveDocument.Save

' bring up filesave and default to "AutoCorrect Backup Document" as the name

```

With Dialogs(wdDialogFileSaveAs)
    .Name = "AutoCorrect Backup Document"
    .Show
    'ACFileName = .Name
End With
'
SaveACDocErrors:
Select Case Err.Number
Case 0:
    ' no error
Case 4198:
    ' cancel
SaveACDoc = False
Case Else
    ' put error number and description in title of the message box
    Title = Err.Number & " " & Err.Description
    Response = MsgBox("Fehler! Wollen Sie es noch einmal probieren?!", _
        vbYesNo + vbCritical + vbDefaultButton2, Title)
If Response = vbYes Then
    Resume 'bring up SaveAs again
Else ' User choose No.
    SaveACDoc = False
End If
End Select
End Function
'-----

```

```

Public Function OpenACDoc(ByVal ACFileOpenName As String) As Boolean
Dim Style As Integer
'
    OpenACDoc = True
    Err.Clear
    On Error GoTo OpenACDocErrors
    Documents.Open FileName:=ACFileOpenName
Exit Function
OpenACDocErrors:
If Err.Number <> 0 Then
    OpenACDoc = False
End If
End Function
'-----

```

```

Sub acl_Dateien_suchen_und_auflisten_und_erklären()
    ' H. Chris Gast, 13. Nov. 2012, Stand 16.Dezember 2012.
    ' Word 2000 bis Word 2010, getestet Windows XP und Windows 7.
    ' Sucht die für die AutoKorrektur nötigen ".acl"-Dateien und gibt ggf. die zugehörige Sprache an.
    ' zum Beispiel "MSO1031.acl Deutsch (Deutschland)".
Dim Pfad1 As String
Dim Name1 As String
Dim Rng0 As Range
Dim i As Integer: i = 0
Dim Abfrage As Integer
    ' *****
    On Error Resume Next
    Pfad1 = Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
    ' Erste acl-Datei abrufen.
    Name1 = Dir(Pfad1 & Application.PathSeparator & "*.acl")
If Err.Number > 0 Or Name1 = "" Then
    MsgBox "acl-Dateien nicht gefunden!" & vbCrLf _

```

```

        & "Versuchen Sie es im Explorer unter:" & vbCrLf _
        & Replace(Options.DefaultFilePath(wdUserTemplatesPath), "vorlagen", "Office")
    Exit Sub
End If
' *****
' Neues Dokument erstellen:
Application.Documents.Add
Selection.WholeStory
Selection.Collapse Direction:=wdCollapseEnd ' (Briefkopf mit Datum soll ggf. erhalten bleiben)
Selection.TypeParagraph
Selection.TypeParagraph
Set Rng0 = Selection.Range
' *****
' Schleife über alle ".acl"-Dateien mit abgefangenen Fehlern:
Do While Name1 <> "" ' Schleife beginnen.
    If Name1 = "" Then Exit Do
    Selection.TypeText Text:=vbCrLf & Name1 & " "
    On Error Resume Next
' Den Sprachcode aus "Name1" der Markierung zuordnen:
Selection.LanguageID = Val(Mid(Name1, 4))
' Aus der zugeordneten Sprache den Sprachnamen entnehmen,
' aber nur dann, wenn die Zuordnung wirklich funktioniert hat:
If Selection.LanguageID = Val(Mid(Name1, 4)) Then
    Selection.TypeText Text:=Languages(Selection.LanguageID)
    ' Zum Beispiel ergibt die Zahl "1031": "Deutsch, Deutschland".
Else
    Selection.TypeText Text:="((Sprache ließ sich nicht aufrufen!))"
End If
If Err.Number = 5941 Then
    If Mid(Name1, 4, 4) = 127 Then
        Selection.TypeText Text:=""Mathe-AutoKorrektur-Liste""
    Else
        Selection.TypeText Text:="((Dieser Nummer ist keine Sprache zugeordnet))"
    End If
    On Error GoTo 0
Elseif Err.Number > 0 Then
    Selection.TypeText Text:=vbCrLf & "unklarer Fehler " & Err.Number
    On Error GoTo 0
End If
i = i + 1
' nächsten Namen abfragen:
Name1 = Dir
Loop
' *****
' Wieder als Sprache "Deutsch" einstellen und Hinweise geben:
Selection.LanguageID = 1031
Selection.TypeParagraph
Selection.TypeParagraph
Selection.TypeText Text:="Sie können die AutoKorrektur-Dateien (*.acl) " _
    & "auf einen USB-Stick speichern und in einen neuen Computer übertragen." _
    & vbCrLf & vbCrLf _
    & "ACHTUNG: AutoKorrekturen mit Bildern, Sonderzeichen oder Formatierungen werden " _
    & "nicht hier, sondern Sprach-unabhängig in der Normal.dot gespeichert."
' *****
' An den Anfang des Dokuments gehen und Zählung ausgeben:
Rng0.Select
Selection.ParagraphFormat.Alignment = wdAlignParagraphLeft
Selection.TypeText Text:"Es gibt im Ordner " & vbCrLf & """" & Pfad1 & """" & vbCrLf & "insgesamt " _
    & i & " AutoKorrektur-Dateien mit der Endung *.acl:" & vbCrLf & vbCrLf
' *****
' Abfrage, ob der Explorer-Ordner mit den ".acl"-Dateien geöffnet werden soll:
Abfrage = MsgBox("Es gibt im Ordner """" & Pfad1 & """" insgesamt " & i & " AutoKorrektur-Dateien." _
    & vbCrLf & vbCrLf _

```

```
        & "Wollen Sie diesen Explorer-Ordner öffnen?", vbYesNoCancel + vbDefaultButton2)
    ' Bei "Ja" kommt "6" heraus, bei "Nein" kommt "7" heraus, bei Abbrechen eine "2".
    If Abfrage = 6 Then
        Shell "Explorer.exe /e, " & Pfad1, vbNormalFocus
    End If
End Sub
'-----
```

Anhang C

Anleitung: **Eigene Buttons** in der Menüleiste (Ribbon) oder in der Schnellstartleiste (QAT) von **Word 2010** einfügen

C.1 Allgemeine Anleitung

In das Menüband von Word 2010 oben am Bildschirm mit der rechten Maustaste klicken und "**Menüband anpassen...**" oder "**Symbolleiste für den Schnellzugriff anpassen...**" wählen. (*Menüband = englisch "Ribbon", Schnellstartleiste = englisch "Quick Access Toolbar", QAT*). Es erscheint folgendes Menü, siehe das folgende Bild C.1.

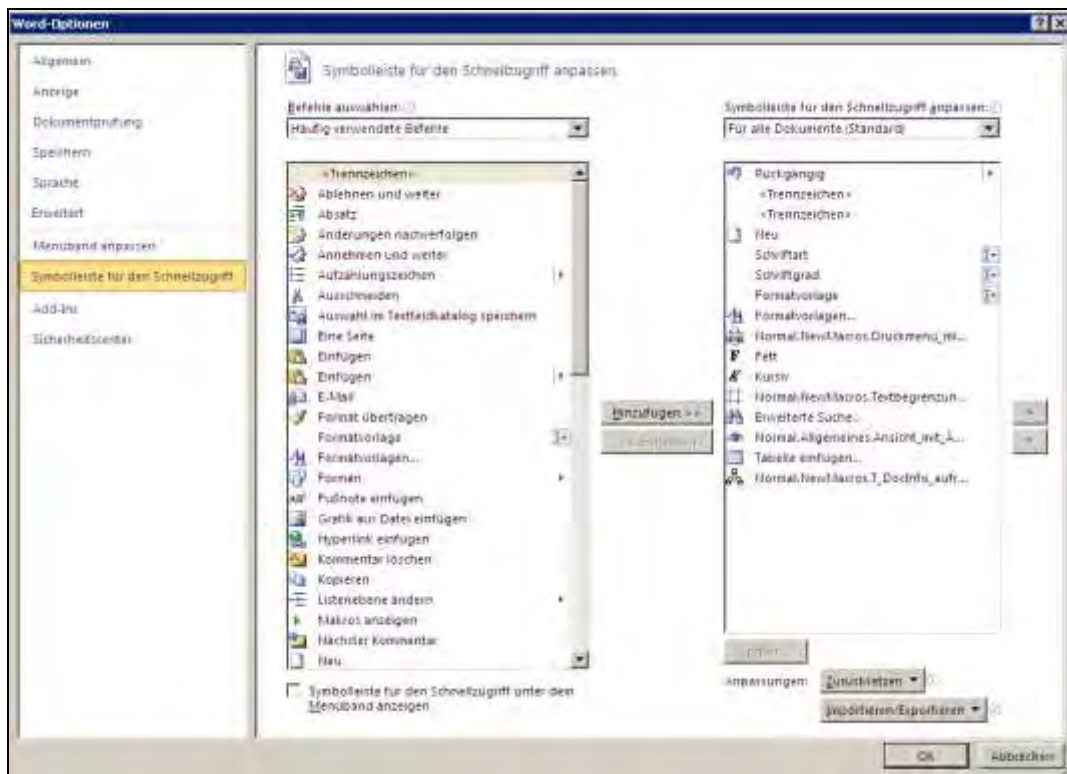


Bild C.1: Anpassen der Menüleiste bzw. der Schnellstartleiste von Word 2010

Links im Bild C.1 ist unter "**Befehle auswählen**" das Makro oder der Befehl auszuwählen und nach rechts unter "**Symbolleiste für den Schnellzugriff**" bzw. "**Menüband anpassen**" hinzuzufügen. Rechts ist die Stelle zu wählen, wo ein neuer Button eingefügt werden soll. Im Falle des Menüband ist dazu ggf. eine "**Neue Registerkarte**" (Ribbon-Tab) und grundsätzlich eine neue "**Gruppe**" (group) anzulegen (nicht im Bild). Anschließend sollte man dann rechts das Eingefügte markieren und über den Button "**Ändern**" eines der vielen Bildzeichen auswählen und bei Menübändern auch den angezeigten Text etwas kürzen. Insgesamt passen in eine Registerkarte (Ribbon-Tab) etwa so viele Buttons wie in drei Symbolleisten (CommandBars) bei früheren Word-Versionen.

Einen "**Schaltflächen-Editor**" zum Bearbeiten der Bildzeichen bzw. zum Neu-Malen gibt es in Word 2010 **leider** nicht mehr ☹.

Die Änderungen werden nicht in der Normal.dotm, sondern in einem eigenen Explorer-Ordner als Datei "**Word.officeUI**" gespeichert. Praktischer ist aber, man klickt im Bild C.1 rechts unten auf "**Importieren/Exportieren**" und wählt dann das Gewünschte. So kann man sich

bequem seine Änderungen speichern ("*Word-Anpassungen.exportedUI*") und ggf. auf einen anderen, neuen Computer übertragen.

C.2 Buttons in der Schnellstartleiste einer Dokumentvorlage (".dotm"-Datei), die nicht die "Normal.dotm" ist

Bei der Schnellstartleiste kann man einige Buttons auch in einer ".dotm"-Datei unterbringen. So kann man Dokumentvorlagen mit eigenen Buttons in der Schnellstartleiste erstellen, die nur dann erscheinen, wenn die geöffnete Datei diese ".dotm"-Datei als Dokumentvorlage hat.

Dazu muss diese Dokumentvorlage für Word 2010 mit der Option "nicht kompatibel mit alten Word-Versionen" erstellt worden sein. Man öffnet sie dann, wählt "Symbolleiste für den Schnellzugriff anpassen..." (im Bild C.1 oben rechts), aber anstatt "Für alle Dokumente (Standard)" wählt man den Namen dieser ".dotm"-Datei. Für Buttons in den Menüleisten (Ribbons) geht dies leider nicht so ohne weiteres, siehe folgenden Abschnitt.

C.3 Buttons in der Menüleiste einer ".dotm"-Datei (Quelltext-Bearbeitung)

Will man in einer Dokumentvorlage auch spezielle Button in den Menüleisten (Ribbon-Tabs) unterbringen, ist dies **ohne** Zusatz-Programme (aus dem Internet) **äußerst** schwierig. Man muss dazu den **Quelltext** bearbeiten, und darf dabei keine Tippfehler begehen.

C.3.1 Lösungsweg mit Beispielen

Ich habe zuerst konventionell ein paar zusätzliche Buttons für die Schnellstartleiste in meiner ".dotm"-Datei erzeugt. Dann hängte ich an den Dateinamen der ".dotm" einfach ".zip" an, und fand dann beim Öffnen mit WinZip darin eine Datei namens "**customUI.xml**", siehe folgendes Bild:

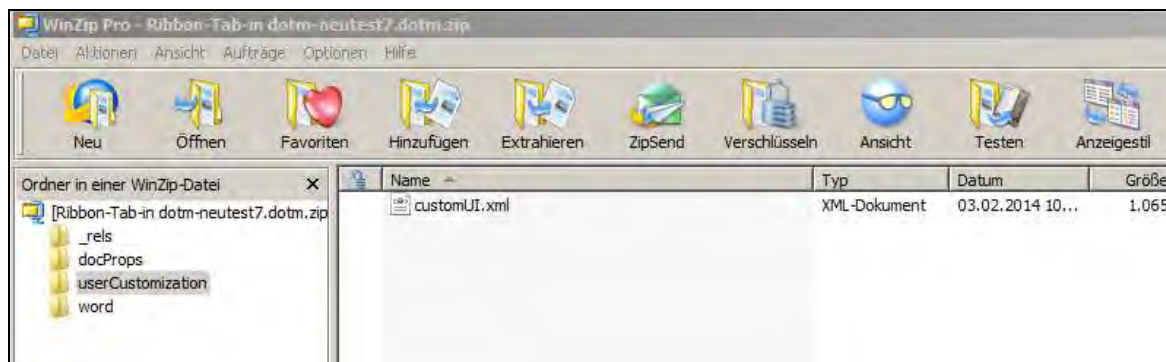


Bild C.2: Screenshot der ".dotm.zip"-Datei mit der darin enthaltenen "customUI.xml"

Nach Öffnen dieser Datei sehe ich einen Text wie:

```
mso:customUI xmlns:x1="PDFMaker.OfficeAddin"
xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui"><mso:ribbon>
<mso:qat><mso:documentControls><mso:control idQ="x1:ACROBAT_CREATE"
visible="true"/></mso:documentControls></mso:qat></mso:ribbon></mso:customUI>
```

Zur Veranschaulichung habe ich Absatzmarken und Farbmankierungen eingefügt. Hier einige Beispiele, die meinen Lösungsweg zum fertigen Ribbon-Tab veranschaulichen. Die Absatzmarken (Zeilenumbrüche) und Leerzeichen dienen hier der besseren Lesbarkeit, stören aber nicht die spätere Funktion. Die Einfärbung der folgenden Beispiele ist in der

"customUI.xml" natürlich nicht enthalten. Das Beispiel 1 sieht dann wie folgt aus (zwecks Übersichtlichkeit hier nur mit einem zusätzlichen Button in der Schnellstartleiste):

Beispiel 1: Quelltext der "customUI.xml" mit einem mittels Word erzeugten Button in der Schnellstartleiste, hier ein Button für die Pdf-Erzeugung:

```
<mso:customUI
  xmlns:x1="PDFMaker.OfficeAddin"
  xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
<mso:ribbon>
<mso:qat>
<mso:documentControls>
<mso:control idQ="x1:ACROBAT_CREATE" visible="true"/>
</mso:documentControls>
</mso:qat>
</mso:ribbon>
</mso:customUI>
```

Hier fügte ich nun mittels Notepad von Hand in der "customUI.xml" an die Stelle des Quelltextes für die Schnellstartleiste von Beispiel 1 die passenden Programmzeilen für ein zusätzliches Ribbon-Tab einfügen, siehe **Beispiel 2** und folgende.

Beispiel 2: Quelltext der von Hand bearbeiteten "customUI.xml" einer ".dotm"-Datei mit einem Ribbon-Tab, erstellt nach vielem Probieren analog zu Mustern aus dem Internet:

```
<customUI
  xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon startFromScratch="false">
<qat>
</qat>
<!-- von Hand erstelltes Beispiel -->
<tabs>
<tab id="Neuer_Tab" label="Neutab" insertBeforeMso="TabHome">
  <group id="Zeichen" label="Zeichen">
    <button id="TestC1" label="M1" imageMso="ListMacros" onAction="Makrobeispiel"/>
  </group>
</tab>
</tabs>
</ribbon>
</customUI>
```

Das Makro braucht nicht in der betreffenden ".dotm"-Datei untergebracht zu sein, wichtig ist nur, dass nicht zwei Prozeduren den gleichen Namen haben:

```
Sub Makrobeispiel()
  MsgBox "Hallo, es geht!"
End Sub
```

Das obige Beispiel 2 erhielt diese Form erst nach einigem Probieren und Fehlersuche.

C.3.2 Fehlersuche

Dieses xml für die Buttons im Ribbon ist sehr empfindlich gegen Fehler. Normalerweise bleiben neue Ribbon-Tabs in separaten ".dotm"-Dateien solange völlig unsichtbar, bis der letzte größere Fehler beseitigt ist. Dabei helfen folgende drei Punkte.

a) In Word 2010 die Option aktivieren:

"Fehler des Benutzeroberflächen-Add-Ins anzeigen".

b) Problem mit typografischen Anführungszeichen:

Es dürfen keinerlei typografischen Anführungszeichen verwendet werden.

c) Probleme mit Umlauten:

Bei Umlauten gibt es leicht Fehler, wenn die "customUI.xml" falsch codiert ist, siehe folgendes Bild.

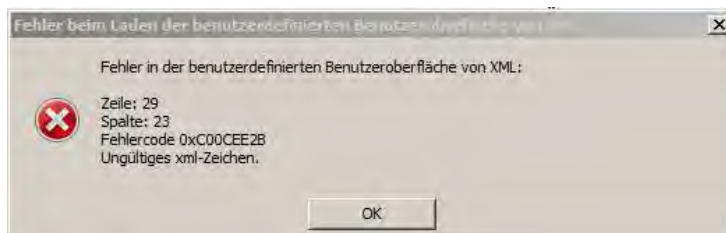


Bild C.3: Fehlermeldung von Word 2010 bei falsch codierten Umlauten im Ribbon-Tab

Lösung: Man bearbeitet die "customUI.xml" nur mit Notepad ("Editor"). So konnte ich nach langer Fehlersuche schließlich auch Umlaute verwenden, ohne dass mir Word die geforderte Codierung ("UTF-8") zerstörte oder typografische Anführungszeichen erzeugte. Die Funktion "Speichern unter" mit Dateityp "Textdatei" und Angabe der Codierung war bei meinem Notepad überflüssig.

Schließlich funktionierte es, siehe folgendes Bild.

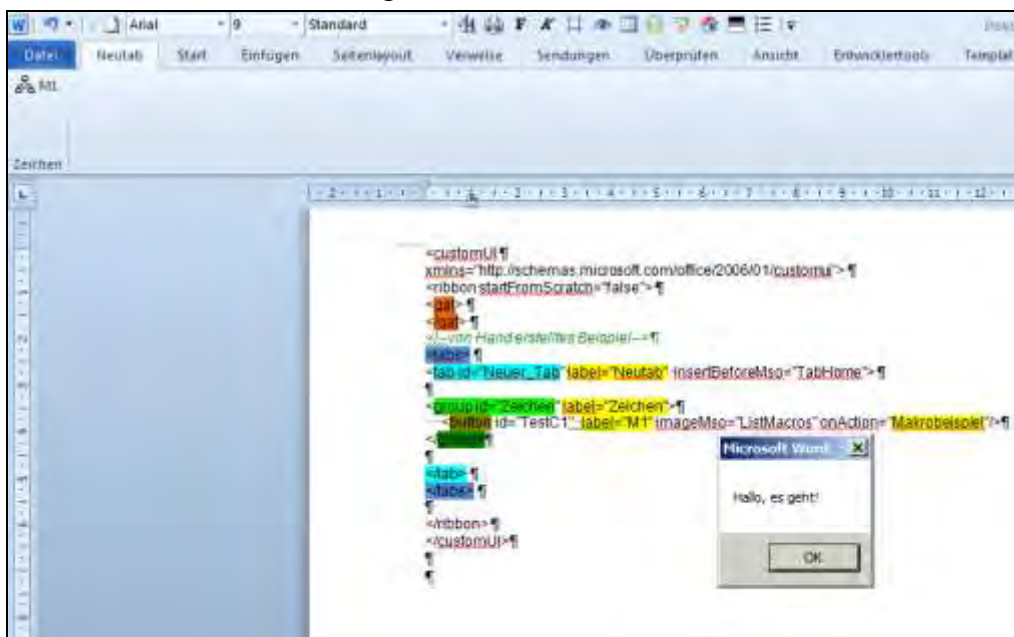


Bild C.4: Screenshot des Beispiels 2

Als nächstes versuchte ich nun, sowohl einen Button in der Schnellstartleiste (QAT) als auch Buttons in einer neuen Registerkarte (Ribbon-Tab) des Menübandes in einer ".dotm"-Datei einzufügen.

Das folgende Beispiel 3 enthält sowohl einen zusätzlichen Button in der Schnellstartleiste als auch drei Buttons in einer neuen Menüleiste (Ribbon-Tab); einen Button für den Befehl "Seitenwechsel oberhalb", einen Button mit dem Makro aus Beispiel 2 und einen Button für die Erzeugung einer Pdf-Datei. Diese Beispiele hatte ich zuerst konventionell (siehe Abschnitt C.1) erzeugt und dann aus der "Word-Anpassungen.exportedUI" rauskopiert.¹⁸

Beispiel 3: Erfolgreiche Kombination von Button in der Schnellstartleiste (QAT) und in einer neuen Registerkarte (Ribbon-Tab) des Menübands

```
<mso:customUI
  xmlns:x1="PDFMaker.OfficeAddin"
  xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
<mso:ribbon>
<mso:qat>
  <mso:documentControls>
    <mso:control idQ="x1:ACROBAT_CREATE" visible="true"/>
  </mso:documentControls>
</mso:qat>
<!-- von Hand erstellt, es geht endlich-->
<mso:tabs>
<mso:tab id="NeuTab" label="Neutest" insertBeforeMso="TabHome">
  <mso:group id="T_Befehle" label="Befehle">
    <mso:button idMso="BreakParagraphPageBreakBefore" label="Seitenumbruch" imageMso="GreenBall"/>
    <mso:button id="TestC1" label="M1" imageMso="ListMacros" onAction="Makrobeispiel"/>
    <mso:control idQ="x1:ACROBAT_CreatePDF"/>
  </mso:group>
</mso:tab>
</mso:tabs>

</mso:ribbon>
</mso:customUI>
```

Damit dies funktioniert, musste ich (nach langem Probieren) fast überall "mso:" davorsetzen, jedoch bei "id" nur teilweise! Beispiel 2 erwies sich hier für mich als Sackgasse.

Mit "insertBeforeMso="TabHome" änderte ich den Einfüge-Ort der neuen "Menüband-Registerkarte" ("Ribbon-Tab"), so dass er links von "Start" erscheint, siehe folgendes Bild (Namen der übrigen Ribbon-Tabs siehe weiter unten Tabelle C.1).

¹⁸ In den "Word-Anpassungen.exportedUI" hieß der Befehl:

```
<mso:control idQ="mso:BreakParagraphPageBreakBefore" label="Seitenumbruch" imageMso="GreenBall" visible="true"/>
```

hiervon ließ ich visible="true" weg.

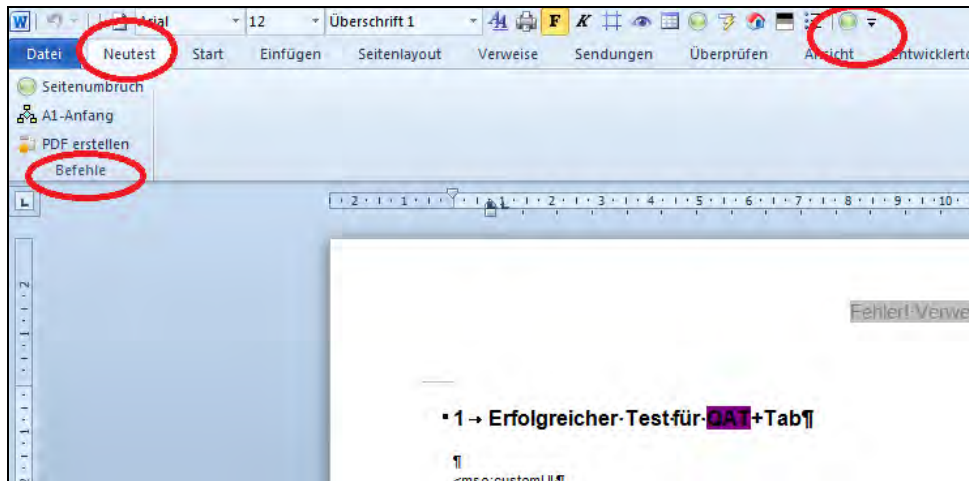


Bild C.5: Erfolgreiche Kombination von Button in der Schnellstartleiste und im Menüband

Man kann auch zusätzliche Buttons in bereits vorhandenen Registerkarten der Menüleiste unterbringen, dies muss aber in einer neuen "Gruppe" erfolgen. Normalerweise enthält ein Ribbon-Tab mehrere Gruppen, siehe Beispiel 4.

Beispiel 4: Kombination von QAT-Befehl, neuem Ribbon-Tab und Ergänzung bei "Start"

```
<?xml version="1.0" encoding="UTF-8"?>
<mso:customUI
  xmlns:x1="PDFMaker.OfficeAddin"
  xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
<mso:ribbon>

<mso:qat>
  <mso:documentControls>
    <mso:control idQ="x1:ACROBAT_CREATE" visible="true"/>
  </mso:documentControls>
</mso:qat>

<!-- von Hand erstellt, Test, 7. März 2014-->
<mso:tabs>
<mso:tab id="NeuTab" label="Neutest" insertBeforeMso="TabHome">
  <mso:group id="T_Befehle" label="Befehle">
    <mso:button idMso="BreakParagraphPageBreakBefore" label="Seitenumbruch" imageMso="GreenBall"/>
    <mso:button id="Test1a" label="Makrotest" imageMso="ListMacros" onAction="Makrobeispiel"/>
    <mso:control idQ="x1:ACROBAT_CreatePDF"/>
  </mso:group>
  <mso:group id="T_Aufz" label="Aufz.">
    <mso:button id="T3a" label="Aufzählung-abc" imageMso="DataGraphicIconSet" onAction="Aufz_Buchst"/>
    <mso:button id="T3b" label="Aufzählung-123" imageMso="DataGraphicIconSet" onAction="Aufz_Ziffern"/>
    <mso:button id="T3c" label="Aufzählung---" imageMso="DataGraphicIconSet" onAction="Aufz_Striche" />
    <mso:button id="T3d" label="Aufzählung-röm" imageMso="DataGraphicIconSet" onAction="Aufz_römisch"/>
  </mso:group>
</mso:tab>

<mso:tab idMso="TabHome">
  <mso:group id="Testgruppe1" label="Sondergruppe">
    <mso:button id="Test1b" label="M1" imageMso="ListMacros" onAction="Makrobeispiel"/>
  </mso:group>
</mso:tab>

</mso:tabs>
</mso:ribbon>
</mso:customUI>
```

Ergebnis

Das Testmakro funktioniert einwandfrei in 2 unterschiedlichen Ribbon-Tabs. Wichtig dabei war, dass der "id"-Name unterschiedlich war (id="Test1a" und id="Test1b").

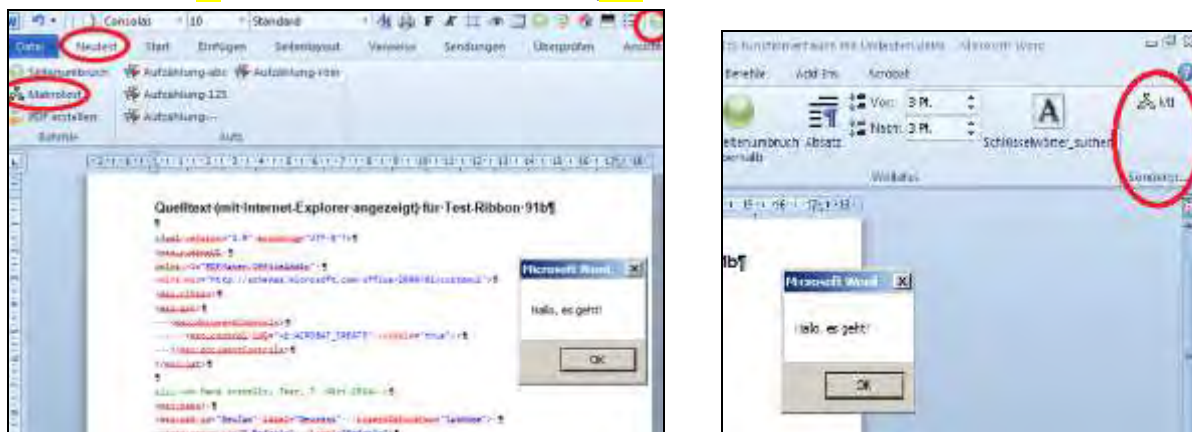


Bild C.6: Screenshot des Beispiel 4 mit Buttons in QAT und zwei Ribbon-Tab

C.3.3 Problem: Buttons funktionieren für Befehle, aber nicht für Makros

Beim Ergänzen einiger Buttons in einer fremden Dokumentvorlage (".dotm") funktionierten bei mir zwar die Buttons für normale Word-Befehle, aber nicht die Buttons für eigene Makros:

```
<group id="Ergaenzung" label="Zusaetzhche_Buttens">  
  <button id="A11a" label="Testmakro" imageMso="ListMacros" onAction="TestMakro"/>  
  <button idMso="BreakParagraphPageBreakBefore" label="Umbruch" imageMso="GreenBall"/>  
</group>
```

Bei dem Testmakro kam die Fehlermeldung:

"Falsche Anzahl an Argumenten oder ungültige Zuweisung zu einer Eigenschaft."

Lösung:

Es fehlte ein sogenannter "*Callback for customButton onAction*"

```
Sub TestMakro(control As IRibbonControl)  
  MsgBox "Hallo, es funktioniert jetzt!"  
End Sub
```

Damit funktionierte es. Bei meiner selbstgestrickten Dokumentvorlage in C.3.1 war dies allerdings nicht nötig.

Ferner zeigte sich bei mir wieder, dass die Verwendung von Umlauten wegen der UTF8-Codierung beim Bearbeiten und Speichern des Quelltextes sehr riskant ist.

C.4 Vorhandene Ribbon-Buttons in einer Dokumentvorlage ausblenden und ersetzen (Quelltext-Bearbeitung)

Will man in allen mit einer speziellen Dokumentvorlage erzeugten Dateien Buttons deaktivieren, die in der Normal.dotm eigentlich vorhanden sind, aber nicht in anderen Dateien, ist dies mit etwas Geschick ebenfalls machbar. Hier als Beispiel das Ersetzen einer ganzen Menüband-Registerkarten-Gruppe.

Zuerst mache ich mir eine Sicherheitskopie des Menübandes (Ribbon) der Normal.dot (siehe Abschnitt C.1) als "*Word-Anpassungen.exportedUI*". Dann erzeuge ich wie dort die gewünschte Änderung (im folgenden Beispiel werden die Buttons für das Einfügen von Bildern ersetzt), exportiere aus dieser geänderten Vorlage die geänderte "*Word-Anpassungen.exportedUI*" und importiere dafür wieder die ursprünglich "*Word-Anpassungen.exportedUI*".

Beim Vergleich der beiden Dateien sehe ich den Unterschied. Habe ich den Button für "Einfügen-Illustrationen" **deaktiviert**, ist der ergänzte Text zum Beispiel wie folgt:

```
<tab idQ="TabInsert">
  <group idQ="GroupInsertIllustrations" visible="false"/>
</tab>
```

Bild C.7: Abweichender Quelltext, mit dem Internet-Explorer, "Ansicht → Quellcode"

Diese Änderung baue ich dann in den Quelltext der Menüleiste meiner speziellen Dokumentvorlage ein. Hier habe ich anstelle der ausgeblendeten Befehle ein Makro aufgenommen, welches anstelle des ausgeblendeten Buttons dafür sorgt, dass hier nur "jpg", "png" und "tif"-Bilder verlinkt werden können.

Zu Testzwecken habe ich hier auch noch den Word-Befehl "Seitenumbruch oberhalb" ergänzt.

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" onLoad="OnRibbonLoad">
<ribbon startFromScratch="false">
<tabs>

<!-- Ergaenzung -->
<tab idQ="TabInsert">
  <group id="a22Verlinken" label="Bild einlinken" insertBeforeMso="GroupInsertIllustrations">
    <button idMso="BreakParagraphPageBreakBefore" label="Seitenumbruch" imageMso="GreenBall"/>
    <button id="A22BildEinlinken" label="Grafik_verlinken" imageMso="BlackAndWhiteInverseGrayscale"
      onAction="Makro_Bildverlinken"/>
  </group>
  <group idQ="GroupInsertIllustrations" visible="false"/>
</tab>
<!-- Ende der Ergaenzung -->

</tabs>
</ribbon>
</customUI>
```

Bild C.8: Neuer Quelltext, "Internet-Explorer → Ansicht → Quellcode"

Ergebnis:

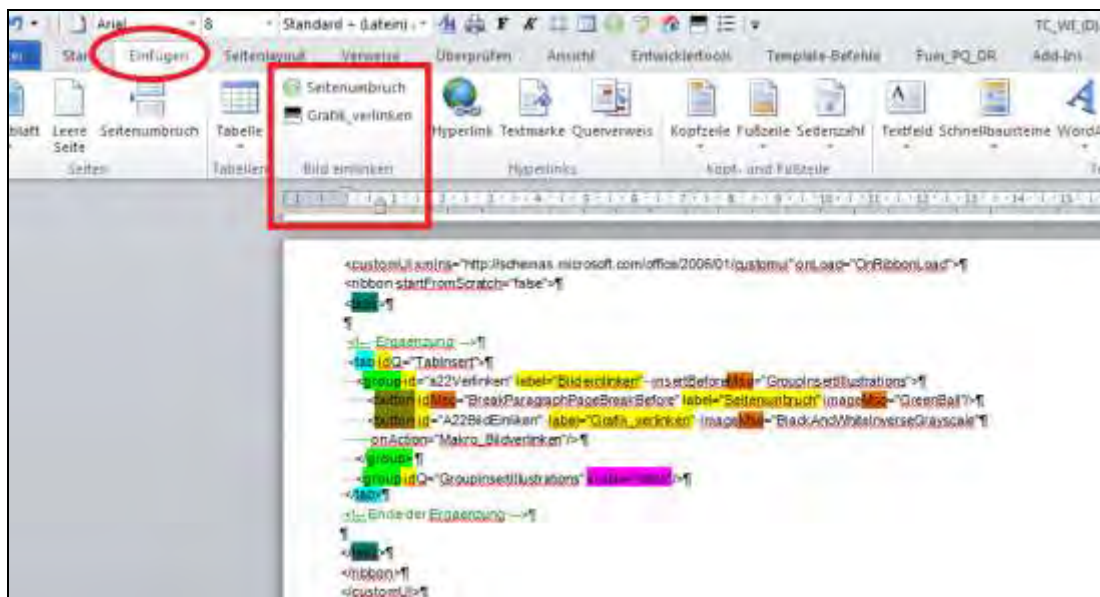


Bild C.9: Ersetzter Button für "Grafik" im Ribbon-Tab "Einfügen"

Makro-Auszug:

```

Sub Makro_Bildverlinken (control As IRibbonControl)
Dim VarFullname As Variant
Dim strFilename As String
Dim ...
" usw.
"
' Bild-Datei einlesen mit Filterung (jpg, png oder tif):
With Application.FileDialog(msoFileDialogOpen) ' neue Funktion von Word 2010
.Title = "Bildauswahl"
.ButtonName = "Auswählen"
.AllowMultiSelect = False
.InitialFileName = "*.tif; *.jpg; *.png"
.Filters.Clear
If .Show = -1 Then
VarFullname = .SelectedItems(1)
strFullname = CStr(VarFullname) ' Pfad + Dateiname
strAnvFile = Dir(VarFullname) ' Dateiname ohne Pfad
strAnvPath = Replace(strFullname, strAnvFile, "") ' nur Pfad
Else
MsgBox "Es wurde nichts ausgewählt!"
Exit Sub
End If
End With
"... usw.
"...
Selection.InlineShapes.AddPicture _
FileName:=strFilename, _
LinkToFile:=True, _
SaveWithDocument:=False
...
End Sub

```

C.5 Tastenkürzel für Buttons und auch für Ribbon-Tabs zuordnen

Erfahrungen:

Es gibt **keinen** Unterschied zwischen Groß- und Kleinbuchstaben bei dem "keytip"-Befehl.

Viele Buchstaben sind bereits fest zugeordnet und funktionieren nicht, so wird etwa bei "n" die Registerkarte (Ribbon-Tab) "Add-Ins" aufgerufen. Es funktionieren die einfache Buchstaben "m", "o" und "q". Es funktioniert auch `keytip="mm"` aber nicht `keytip="%MM"`

Es dürfen **keinerlei** Namen doppelt vorkommen, außer bei "idQ=" und "idMso=".

Bei Umlauten darf man keine Texte aus einer Word-Datei reinkopieren, oder man muss neu "Speichern unter" mit Format "UTF-8". Auch Umlaute in der Kommentarzeile sind riskant.

Tastenkürzel für die Schnellstartleiste (QAT) sind mir nicht gelungen. Aber man kann natürlich allen Befehlen und Makros Tastenkürzel zuweisen, unabhängig davon, ob sie einen Button in der Schnellstartleiste oder in der Menüleiste (Ribbon) haben, siehe [1].

Der Quellcode eines Befehls in der Schnellstartleiste (QAT) unterscheidet sich von dem in einer Menüband-Registerkarte (Ribbon-Tab).

In Computern, bei denen kein Adobe X installiert ist, sind die zugehörigen Buttons unsichtbar. Ich habe den Button für Pdf-Erzeugung hier als Beispiel hinzugefügt, weil es für mich das schwierigste von allem war.

Die "Custom.xml" mit **Tastenkürzeln** in der Datei "Ribbontest.dotm"

```
<mso:customUI
xmlns:x1="PDFMaker.OfficeAddin"
xmlns:mso="http://schemas.microsoft.com/office/2006/01/customui">
<mso:ribbon>
<mso:qat>
<mso:documentControls>
<mso:control idQ="x1:ACROBAT_CREATE" visible="true" imageMso="HappyFace"/>
<mso:control idQ="mso:BreakParagraphPageBreakBefore" visible="true"/>
</mso:documentControls>
</mso:qat>

<!-- Erfolgreicher Keytip-Test für Tab, 18.3.2014-->
<mso:tabs>
<mso:tab id="TestTab1" label="Testtab1" keytip="Q" insertBeforeMso="TabHome">
<mso:group id="T_Befehle" label="Befehle">
<mso:button id="TA1" label="Testmakro" imageMso="ListMacros" keytip="K1" onAction="Makrobeispiel" />
<mso:control idQ="x1:ACROBAT_CreatePDF" keytip="K2" />
<mso:button idMso="BreakParagraphPageBreakBefore" keytip="K3" imageMso="GreenBall" />
</mso:group>
</mso:tab>

<mso:tab id="TestTab2" label="Testtab2" keytip="MM" insertBeforeMso="TabHome">
<mso:group id="T_Befehle2" label="dieselben Befehle">
<mso:button id="TA2" label="Testmakro" imageMso="ListMacros" keytip="K1" onAction="Makrobeispiel" />
<mso:control idQ="x1:ACROBAT_CreatePDF" keytip="K2" />
<mso:button idMso="BreakParagraphPageBreakBefore" keytip="K3" imageMso="GreenBall" />
</mso:group>
</mso:tab>

</mso:tabs>
</mso:ribbon>
</mso:customUI>
```

Das folgende Bild zeigt einen Screenshot der Quelltext-Ansicht mit dem Internet-Explorer.

```

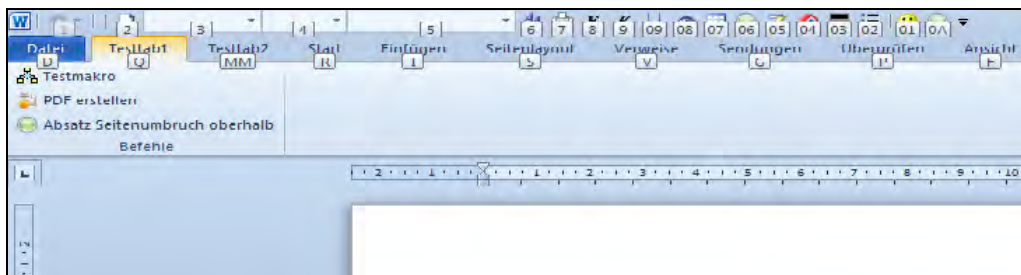
1 <ms:customUI
2 <ms:xmlns:x1="PDFMaker.OfficeAddin"
3 <ms:xmlns:res="http://schemas.microsoft.com/office/2006/01/customui">
4 <ms:ribbon>
5 <ms:ig>
6 <ms:documentControls>
7 <ms:control id="x1:ACROBAT_CREATE" visible="true" imageMso="HappyFace"/>
8 <ms:control id="x1:BreakParagraphPageBreakBefore" visible="true"/>
9 </ms:documentControls>
10 </ms:ig>
11
12 <!-- Erfolgreicher Keytip-Test für Tab, 18.3.2014 -->
13 <ms:tabs>
14 <ms:tab id="TestTab1" label="Testtab1" keytip="Q" insertBeforeMso="TabMose">
15 <ms:group id="1_Befehle" label="Befehle">
16 <ms:button id="TA1" label="Testmakro" imageMso="ListMacros" keytip="K1" onAction="Makrobeispiel" />
17 <ms:control id="x1:ACROBAT_CreatePDF" keytip="K2" />
18 <ms:button id="x1:BreakParagraphPageBreakBefore" keytip="K3" imageMso="GreenBall" />
19 </ms:group>
20 </ms:tab>
21
22 <ms:tab id="TestTab2" label="Testtab2" keytip="W" insertBeforeMso="TabMose">
23 <ms:group id="2_Befehle2" label="dieselben Befehle">
24 <ms:button id="TA2" label="Testmakro" imageMso="ListMacros" keytip="K1" onAction="Makrobeispiel" />
25 <ms:control id="x1:ACROBAT_CreatePDF" keytip="K2" />
26 <ms:button id="x1:BreakParagraphPageBreakBefore" keytip="K3" imageMso="GreenBall" />
27 </ms:group>
28 </ms:tab>
29
30 </ms:tabs>
31 </ms:ribbon>
32 </ms:customUI>
33
34

```

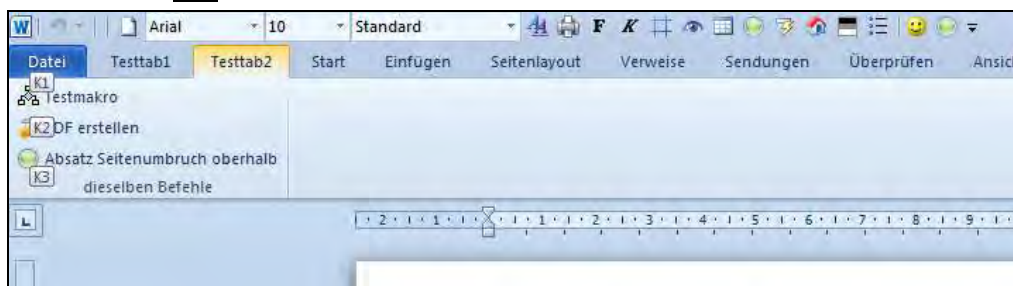
Bild C.10: Screenshot der Quelltext-Ansicht mit dem Internet-Explorer

Screenshots der Tastenkürzel, nach Drücken der **Alt**-Taste

Das folgende Bild zeigt in zwei Screenshots die Verwendung von Tastenkürzeln.



a) nach Drücken von **Alt**



b) nach Drücken von **Alt** und dann **m** und nochmals **m**

Bild C.11: Screenshots für Keytip

Das zugehörige Testmakro:

```
Sub Makrobeispiel()  
  MsgBox "Hallo, es geht!"  
End Sub
```

oder, wenn das Makro nicht in der Makro-Liste beim Aufruf mit **Alt** + **F8** erscheinen soll:

```
Sub Makrobeispiel(Optional Makro_verstecken As Integer)  
  MsgBox "Hallo, es geht!"  
End Sub
```

Für den Aufruf der neuen Registerkarte und eines der darin enthaltenen Buttons mittels SendKeys-Befehl in einem Makro:

```
Sub Aufruftest_mit_Sendkeys()  
  ' H. Chris Gast, 17. März 2014  
  ' SendKeys-Test für Pdf-Erzeugung:  
  ' Der Befehl "SendKeys" funktioniert nur bei Start des VBA-Makros durch Tastenkürzel,  
  ' wenn das Makro vorher eine Pause von 1 Sekunde macht!  
  Dim tt As Single  
  tt = Timer  
  Do While Timer < tt + 1# ' in Sekunden  
    DoEvents  
  Loop  
  SendKeys "(%q)" ' Imitiert Aufruf des Ribbon "Testtab2" mittels Tastenfolge Alt + q.  
  tt = Timer  
  Do While Timer < tt + 0.1 ' kurze Pause  
    DoEvents  
  Loop  
  SendKeys "(k1)" ' Imitiert Tastenfolge "k" und dann die "1".  
End Sub
```

C.6 Buttons in einer vorhandenen Registerkarte (Ribbon-Tab) ergänzen¹⁹

Man kann auch zusätzliche Buttons in einer vorhandenen Registerkarte wie hier unter "Einfügen) zusätzlich unterbringen, und zwar in einer neuen "Gruppe". Diese zusätzliche Gruppe erscheint nur, wenn die ".dotm"-Datei mit dem folgenden Quelltext geöffnet ist bzw. als Dokumentvorlage aktiv ist, sonst nicht.

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
<ribbon>
<tabs>

<!-- Hier vielleicht einige zusaetzliche Registerkarten (Ribbon-Tabs) -->

<tab idMso="TabInsert">
<group id="customGroup" label="AAAA-Test" insertAfterMso="GroupInsertIllustrations">
<button id="customButton" label="Testmakro" imageMso="ListMacros" onAction="a0_Testmakro" />
</group>
</tab>

</tabs>
</ribbon>
</customUI>
```

Dazu gehört hier das Test-Makro:

```
' Makro mit "callback for customButton onAction":
Sub a0_Testmakro(control As IRibbonControl)
MsgBox "Hallo Welt, es geht!"
End Sub
```

Dies ergibt, wenn alles richtig gemacht wurde:

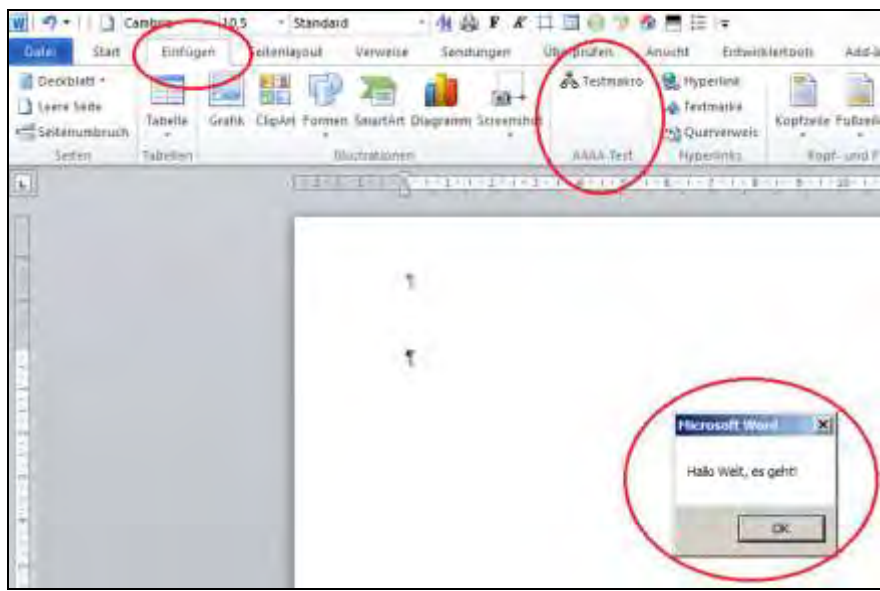


Bild C.12: Screenshot einer "Gruppe" mit einem Button für ein Makro in der Registerkarte "Einfügen"

¹⁹ Aus: <http://www.akademie.de/wissen/eigene-befehle-in-ribbons-einfuegen>, 16. Februar 2015, modifiziert.

Wenn beim Makro der Zusatz "(control As IRibbonControl)" fehlt, kommt hier folgende Fehlermeldung:



Bild C.13: Fehlermeldung, wenn "(control As IRibbonControl)" fehlt

Bei meiner selbstgestrickten Dokumentvorlage in C.3.1 war dieser Zusatz nicht nötig.

C.7 Weitere Tipps, Namen der eingebauten Ribbons und einige Button-Farben

Den Befehl `visible="true"` kann man weglassen, ebenso den Befehl `size="..."`.

Neben dem Einfügen vor (links von) einem Ribbon-Tab

`insertBeforeMso="TabHome"`

gibt es auch das Einfügen hinter (rechts von) einem Ribbon-Tab:

`insertAfterMso="TabView"`

Will man den Quelltext der "Word-Anpassungen.exportedUI" (siehe C.1) nachbearbeiten, kann man dort in einem Ribbon-Tab mit sehr vielen Buttons den Befehl `autoScale="True"` durch `autoScale="false"` ersetzen. Dann werden alle Icons in der jeweiligen Gruppe gleichgroß.

Die folgende Tabelle zeigt die Namen der eingebauten Registerkarten (Ribbon-Tabs) des Menübandes (Ribbon) von Word 2010.

Tabelle C.1: Namen der eingebauten Ribbon-Tabs in Word 2010

Quelltext-Name	Englische Namen (Ribbon-Tabs)	Deutsche Namen (Registerkarten)	Französische Namen (l'onglets)
-	File	Datei	Fichier
TabHome	Home	Start	Accueil
TabInsert	Insert	Einfügen	Insertion
TabPageLayoutWord	Page Layout	Seitenlayout	Mise en page
TabReferences	References	Verweise	Références
TabMailings	Mailings	Sendungen	Publipostage
TabReviewWord	Review	Überprüfen	Révision
TabView	View	Ansicht	Affichage
TabDeveloper	Developer	Entwicklertools	Développeur
TabAddIns	Add-Ins	Add-Ins	Compléments

Die folgende Tabelle zeigt die Namen und die ungefähren Farben einiger Buttons, wie sie bei mir dann in etwa aussehen.

Tabelle C.2: Farben der Buttons "AppointmentColor"

Code	Farbe
imageMso="AppointmentColor0"	weiß
imageMso="AppointmentColor1"	rot
imageMso="AppointmentColor2"	dunkelblau
imageMso="AppointmentColor3"	grün
imageMso="AppointmentColor4"	grau (olive)
imageMso="AppointmentColor5"	orange
imageMso="AppointmentColor6"	hellblau
imageMso="AppointmentColor7"	dunkelgelb/braun
imageMso="AppointmentColor8"	lila
imageMso="AppointmentColor9"	grünblau
imageMso="RecurrenceEdit"	weiß mit dickem Rahmen

C.8 Alte Symbolleisten von Word 2000 usw. in Word 2010 entfernen

Verwende ich eine in Word 2000 oder Word 2003 erstellte Datei bzw. Dateivorlage mit eigenen Symbolleisten (CommandBars) in Word 2010, tauchen diese dort auf der Menüband-Registerkarte "Add-Ins" auf und lassen sich dort nicht ändern, sondern nur als Ganzes löschen.

Direkte Lösung für die Löschung:

Eines der (noch) funktionierenden Icons mit rechter Maustaste anklicken, dann auswählen:

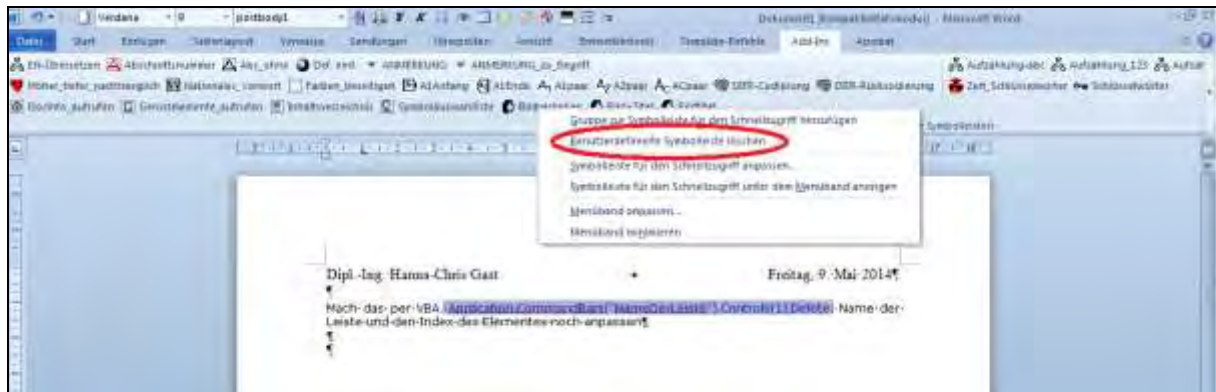


Bild C.14: Screenshot "Benutzerdefinierte Symbolleiste löschen"

Dann erscheint die Warnung:

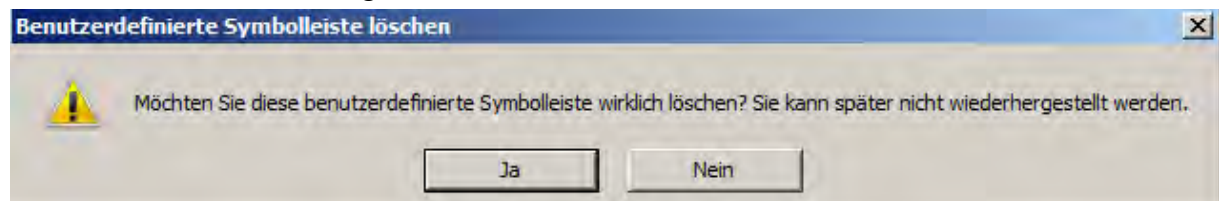


Bild C.15: Screenshot der daraufhin erfolgenden Warnung

Bei mir wird beim Schließen von Word immer noch abgefragt, ob ich die Änderungen in der Dokumentvorlage ("xxx.dot") speichern will. Aber darauf würde ich mich nicht verlassen.

C.9 Vermeintliches Problem "Menüband ist verschwunden"²⁰

Grundsätzlich kann in allen Office 2010 Anwendungen das Menüband (Ribbon) ein- und ausgeblendet werden. Sollten Sie das Menüband unabsichtlich ausgeblendet haben, können Sie es ganz einfach wieder maximieren.

Durch Anklicken des kleinen Pfeils ganz rechts oben können Sie das Menüband ein- oder ausblenden lassen, siehe folgenden Screenshot.

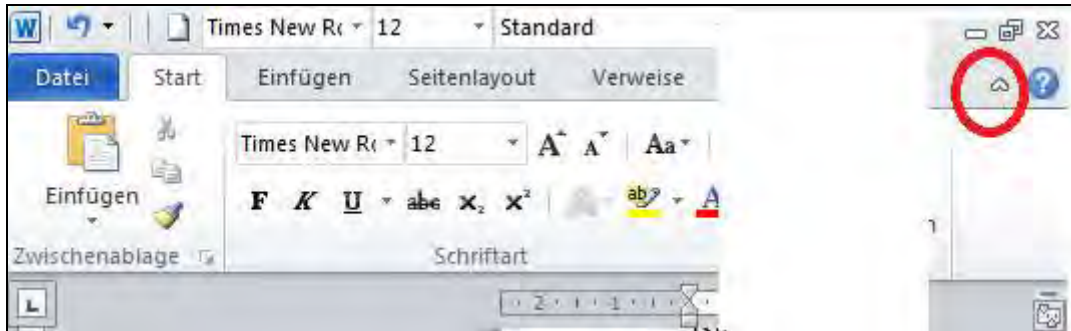


Bild C.16: Menüband ein- und ausblenden

Noch einfacher geht es per Doppelklick auf einen Registernamen im Menüband.

²⁰ [Aus: <http://www.knowhow.de/blog/139/>, B. Deckenbach, 16 Mai 2011, Download: 29. Januar 2016, gekürzt und modifiziert und Bild neu erstellt.

Weitere Literatur

- [1] Hanna-Chris Gast: "**Tastenkombinationen (Tastenkürzel) für Word, Windows, Internet-Explorer und Adobe-Pdf**"; Berlin, Stand 2016, als Download unter: <http://www.siebener-kurier.de/chris-aufsaeetze/Word-Tastaturbefehle.pdf>.
(enthält auch eine Anleitung für die Zuordnung von Tastenkürzeln und Icons).
- [2] Hanna-Chris Gast: "**Schrift- und Sonderzeichen in Word**", Broschüre im Selbstverlag, Stand 2017, im Internet als Download unter: <http://www.siebener-kurier.de/chris-aufsaeetze/Word-Schriftzeichen.pdf>.
- [3] Hanna-Chris Gast: "**Word-Makros für Unicode**, Analyse von Schrift- und Sonderzeichen, Ersatz von Symbol- und Wingdingszeichen durch Unicode sowie ein verbessertes Inhalt-Einfügen"; Selbstverlag; Neu-Auflage, Berlin 2015, Stand 2017; http://www.siebener-kurier.de/chris-aufsaeetze/Unicode_Word-Makros.pdf;
Die Makros aus diesem Aufsatz sind in einer fertigen Dokumentvorlage **downloadbar**, <http://www.siebener-kurier.de/chris-aufsaeetze/Unicode-Makros.zip>.
- [4] Hanna-Chris Gast: "**Einige Makros für Word (und deren Installation)**"; Stand 2017; Download: <http://www.siebener-kurier.de/chris-aufsaeetze/Word-Makros.docx>
- [5] Hanna-Chris Gast: "**Formblatt mit Makros zur Übertragung von Tastenkürzel und AutoKorrekturen für Word**"; Berlin 2011, Stand 2015; **Als Download**: http://www.siebener-kurier.de/chris-aufsaeetze/Formblatt-Tastenkuerzel+Autokorrekturen_uebertragen.doc
- [6] Hanna-Chris Gast: "Ergonomisch eingerichteter Computer-Arbeitsplatz", Stand 2017, www.siebener-kurier.de/chris-aufsaeetze/Ergonomie_fuer_Bildschirmarbeitsplatz.pdf
- [7] Dr. V. Thormählen: "**Mathematische Formeln in Word – Der Formel-Editor in Word 2010**"; Abruf 6. Oktober 2014; http://dr-thormaehlen.de/WORD/Formel_Editor.pdf.
- [8] G. M. Jens: "**Mathematische Formeln in Word 2010**"; Hochschulrechenzentrum Justus-Liebig-Universität Gießen; Abruf zuletzt 6. Nov. 2015, http://dr-thormaehlen.de/WORD/Formel_Editor.pdf
- [9] Holger Matthes: "**Grafiken einfügen**", Abruf 12. August 2015; <http://www.holgermatthes.de/diplom-reader/word/grafiken-einfuegen.php>.
- [10] Wikipedia: "**Mathematical markup language**"; Abruf 15. September 2015; https://en.wikipedia.org/wiki/Mathematical_markup_language.

Impressum

Hanna-Chris Gast:

"Einige Probleme mit Word und deren Lösung"

Selbstverlag, Mai 2012, **Stand**: 11. Oktober 2019

Als Download: <http://www.siebener-kurier.de/chris-aufsaeetze/Word-Probleme-Anleitungen.pdf>

Dipl.-Ing. H. Chris Gast

Bergstr. 27

12169 Berlin

Im Internet: www.siebener-kurier.de/chris

E-Mail: [hcgast "ät" siebener-kurier.de](mailto:hcgast@siebener-kurier.de)

Kopieren mit Quellenangabe erlaubt, Fehlermeldungen und Verbesserungsvorschläge an mich sogar **erwünscht!**