

**ChangeChars**

**COLLABORATORS**

	<i>TITLE :</i> ChangeChars		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 3, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ChangeChars</b>	<b>1</b>
1.1	ChangeChars (c) 1994 Volker Voigt . . . . .	1
1.2	Einleitung . . . . .	1
1.3	Disclaimer . . . . .	2
1.4	Installation . . . . .	2
1.5	Anleitung . . . . .	2
1.6	Eigene Konvertierungstabellen . . . . .	3
1.7	History & Credits . . . . .	4
1.8	Autor . . . . .	4

---

# Chapter 1

# ChangeChars

## 1.1 ChangeChars (c) 1994 Volker Voigt

ChangeChars v1.0  
written by Volker Voigt

Tool zur Text-Konvertierung auf Zeichen-Ebene

Einleitung - Was ist ChangeChars und warum?

Disclaimer - Rechtliches: Bei Benutzung von  
ChangeChars werden die Disclaimer anerkannt.

Installation - Einfacher geht es kaum

Anleitung - Wie nutzt man ChangeChars voll aus?

History & Credits

Autor - Who did it?

## 1.2 Einleitung

---

Motivation:

ChangeChars entstand aus zwei Gründen:

- Nachdem ich mir in einer Box ein knappes MByte Songtexte gesaugt hatte, mußte ich feststellen, dass die Texte im IBM-ASCII-Zeichensatz abgetippt worden waren, was bei Umlauten in meinen Textviewern zu Sonderzeichen führte. Ein Konvertierungsprogramm musste her.
- Da ich erst vor einiger Zeit angefangen habe, mich mit C zu beschäftigen und bisher noch kein "sinnvolles" Utility geschrieben habe, war das genau das Richtige.

(Also nicht zuviel erwarten ;-)

Was es macht:

ChangeChars kann nur von der Shell aus benutzt werden.

ChangeChars braucht neben dem zu konvertierenden File eine Konvertierungstabelle, in der die zu ändernden Zeichen abgelegt sind. Wahlweise wird das Originalfile überschrieben oder ein zweites File erzeugt (sozusagen eine konvertierte Kopie angelegt).

## 1.3 Disclaimer

Es wird keine Garantie auf die Funktionstüchtigkeit dieses Programms gegeben. Die Benutzung erfolgt auf eigene Gefahr. Der Autor kann unter keinen Umständen für Schäden haftbar gemacht werden, die direkt oder indirekt durch die Benutzung des Programms entstehen.

Das Programmpaket ist Freeware, d.h. der Benutzer braucht keine Gebühr bzw. Bezahlung an den Autor entrichten, um es benutzen zu dürfen. Alle Rechte bleiben beim Autor. Das Programm darf nur in unveränderter Form zusammen mit dieser Dokumentation weitergegeben werden. Die kommerzielle Vermarktung als Einzelprodukt oder als "Beigabe" zu einem kommerziellen Produkt darf nur mit schriftlicher Erlaubnis des Autors geschehen. Verbreitung via Disketten, BBSs, AmiNet, CDs, etc. die eine Sammlung von Freeware/PD/Shareware darstellen, ist erlaubt.

## 1.4 Installation

ChangeChars fühlt sich eigentlich überall wohl, besonders im Pfad bzw. in C:.. Die Default-Konvertierungstabelle (ChangeChars.cnv) hat ihre Platz in S:..

## 1.5 Anleitung

Die komplette Befehlssyntax:

```
ChangeChars <InFile> [<OutFile>] [-c<Konv.Tabelle>]
```

Das zu konvertierende File <InFile> kann logischerweise nicht weg gelassen werden.

Gibt man optional ein <OutFile> an, wird die Konvertierung nicht im <InFile> durchgeführt, sondern eine neue (konvertierte) Datei erstellt.

Mit Option -c<Konv.Tabelle> kann man eine alternative Konvertierungstabelle angeben. Läßt man diese Option weg, versucht ChangeChars die Default-Tabelle S:ChangeChars.cnv zu öffnen. ←

Einige

Konvertierungstabellen

sind bereits mitgeliefert. IBM2Amiga.cnv dürfte wohl die gängigste sein. Sie konvertiert die Umlaute und das "sz" vom IBM-Zeichensatz zum Amiga-Zeichensatz. Amiga2IBM.cnv macht das Gegenteil. Gross2Klein.cnv macht aus allen 26+3 Großbuchstaben Kleinbuchstaben (also inklusive Umlauten), der Partner dazu ist Klein2Gross.cnv. KillAccent.cnv macht aus allen àùê (im Amiga-Zeichensatz) handliche "aeiou"s, sowohl die kleinen, wie auch die grossen.

Sinnvollerweise kopiert man die am meisten gebrauchte Tabelle nach S:ChangeChars.cnv, um Option -c einsparen zu können. Bei mir ist das IBM2Amiga.cnv (wofür das Programm eigentlich entstanden ist).

Wie bastelt man eigene Konv.Tabellen?

## 1.6 Eigene Konvertierungstabellen

Die Konvertierungstabellen beschreiben, welche Zeichen wie geändert werden sollen, also in welches andere Zeichen sie konvertiert werden sollen.

Zugegeben, die Lösung ist einfach nur billig, aber ich hatte keine Lust, irgendein raffiniertes Tool dafür zu schreiben :)

Man nehme einen Texteditor und erstelle ein File, das untereinander die zu konvertierenden ASCII-Codes auflistet. Beispiel:

```
148      IBM "ö"
246      Amiga "ö"
129      IBM "ü"
252      Amiga ü"
132
```

```
228      ... etc.  
225  
223  
... etc.
```

Wichtig dabei sind zwei Sachen:

Zuerst der zu konvertierende ASCII-Code, dann der "Ziel"-ASCII-Code, dann wieder ein zu kon... etc. (Daraus ergibt sich, daß die Tabelle eine gerade Anzahl von Codes aufweisen muss :) Nach dem letzten Ziel-Code darf `_keine_` Zeile mehr folgen, auch nicht Leerzeilen.

Die Codes müssen immer dreistellig angegeben werden, also mit führenden Nullen, falls nötig (096).

Folgende Kontruktion geht auch:

```
065      A ...  
066      nach B  
066      B ...  
065      nach A
```

Alle "A"s werden zu "B"s und umgekehrt.

Verschlüsselungen von Texten mit Hilfe einer geeigneten Tabelle sind meiner Meinung nach wenig sicher. Binaries jedoch könnte man durch simples Tauschen einiger Zeichen bestimmt gut sichern. Naja, war nur ein Vorschlag.

## 1.7 History & Credits

History:

```
v1.0          27.12.1994  Erste Veröffentlichung.  
                Bugs: Hoffentlich keine.  
  
v0.1alpha - 24.12.-    Qualen, "Bugfixes", Fluchen, temporäre Euphorien.  
v0.9beta     26.12.94
```

Credits:

Danke an Andre "ABernie" Bernhardt fürs Beta-Testen und fürs Uploaden ins Aminet.

Danke an Mami für die intravenöse Ernährung während der Programmierarbeit.

## 1.8 Autor

Der Autor kann erreicht werden über:

```
FIDO:    2:2476/12.1  
UseNet:  VoVo@pl.aop.schiele-ct.de
```

Snail: Volker Voigt  
Nikolaus-Lenau-Str.1  
76199 Karlsruhe

Abschiedsworte:

NO CARRIER

---