Das $\LaTeX\,2_\varepsilon$ Umlaute-Paket*†

Harald Axel Sommerfeldt
Harald_Sommerfeldt@w.maus.de

16. Februar 1995

Zusammenfassung

Es war einmal ein $\LaTeX\,2_\varepsilon$-Paket namens Umlaute [1], welches die Portabilität von $\LaTeX\,2_\varepsilon$ Dokumenten mit enthaltenen Sonderzeichen (d.h. mit ASCII-Kodes ≥ 128) gewährleistete. Hierbei wurden die Zeichenkodierungen `Atari`, `ISO 8859/1 (Latin1)`, `Apple Macintosh`, `PC codepage 850` und `Roman8` unterstützt.

Heutzutage wird einem hierfür das $\LaTeX\,2_\varepsilon$-Paket Inputenc [2] angeboten, welches neuerdings (genauer: seit dem 1.12.1994) dem $\LaTeX\,2_\varepsilon$ beiliegt. Leider werden jedoch nur Kodetabellen für ASCII und den ISO 8859/1 mitgeliefert. „Diesem Mißstand muß Einhalt geboten werden!“ dachte ich mir und zimmerte die Version 2 dieses Paketes zusammen, dem die schmerzlich vermissten Kodetabellen beiliegen.

1 Einleitende Worte

Jeder deutsche LaTeX-Anwender kennt das Problem mit den Umlauten, an die in Amerika mal wieder keiner gedacht hat, als TeX entstand. Es gab zwar schon die verschiedensten Lösungen, um diese direkt in sein Dokument eintippen zu können, aber fast alle hatten einen sehr großen Nachteil: Inkompatibilität! Man konnte seine Dokumente einfach nicht so weitergeben; die Umlaute sorgten auf fremden Rechnern meist nur für ablehnende Fehlermeldungen. Für einige Kodierungen existierten jedoch auch LaTeX-Styles (wie z.B. `isolatin.sty` für die ISO 8859/1-Kodierung), mit denen man die Sonderzeichen ungehemmt verwenden konnte. Besitzer anderer Rechner konnten solche Dokumente trotz „falscher“ Umlaute auch bei sich problemlos übersetzen, vorausgesetzt auch hier war der passende Style vorhanden. Selbst wenn für den eigenen Rechner kein passender Style vorhanden war — man konnte sich ihn einfach selber stricken (so ist z.B. im Kopka [3] in Abschnitt D.3.6 einer abgedruckt), man durfte aber unter keinen Umständen vergessen, diesen mit weiterzugeben.

Aus diesem Mißstand heraus entstand die Idee, endlich mal alle möglichen Zeichensatzkodierungen unter einen Hut zu bringen und für alle verfügbar zu machen; dies war die Entstehung des Umlaute-Paketes für $\LaTeX\,2_\varepsilon$. Die Auswahl der gewünschten Kodierung konnte hierbei einfach über die Option, die man seit $\LaTeX\,2_\varepsilon$ ja jedem Paket mit auf die Reise geben kann, erfolgen.

* Der Name ist auf historische Gründe zurückzuführen: Die ersten Versionen dieses Paketes unterstützten lediglich die Direkteingabe deutscher Sonderzeichen.

† Dieses Paket besitzt die Versionsnummer 2.0a, zuletzt geändert am 5.2.1995.

Nun, mittlerweile gibt es sogar ein Paket vom LaTeX3-Team selber, welches seit der <1994/12/01>-Version jedem LaTeX 2_ε beiliegt; dieses hört auf den Namen Inputenc. Die Kodetabellen sind hier nicht wie beim alten Umlaute-Paket direkt in den Style mit eingearbeitet, sondern sind extern und werden bei Bedarf geladen. Leider werden z.Z. nur ASCII und ISO 08/15 unterstützt.

Da ich noch nie ein Freund langen Wartens war, habe ich also kurzerhand die Kodetabellen vom Umlaute-Paket in das Format des Inputenc-Paketes umgestrickt und vervollständigt. Schnell noch ein neues `umlaute.sty` zusammengestrickt, welches einfach nur noch `inputenc.sty` mit den passenden Optionen aufruft, und voilá – das Umlaute-Paket Version 2 ward geboren!

2 Und wie benutze ich nun dieses Paket?

Die Installation geht wie bei allen LaTeX 2_ε-Paketen vonstatten:
Man übersetze die Datei `umlaute.ins` mit LaTeX, etwa mit

`latex umlaute.ins` .

Die dadurch erzeugten `.sty` und `.def`-Dateien kopiere man anschließend in ein Verzeichnis, wo LaTeX sie auch später wiederfindet, z.B. nach

`/usr/local/lib/texmf/tex/contrib`

oder nach

`\tex\inputs\contrib` ;

die Pfade sind je nach verwendeter TeX-Implementation recht unterschiedlich. Im Zweifelsfalle kopiere man die Dateien einfach dorthin, wo auch die Datei `inputenc.sty` zu finden ist.

In der Präambel (das ist der Teil vor `\begin{document}`) eigener Dokumente kann man nun folgende Zeile verwenden:

`\usepackage[`*encoding name*`]{inputenc}` ,

wobei für *encoding name* entweder `atari`, `pc850`, `isolatin`, `mac` oder `roman8` einzusetzen ist. Und schon kann man in dem Dokument selber die Umlaute (und sonstige Sonderzeichen) munter direkt verwenden.

In dem Inputenc-Paket existiert auch noch der Befehl

`\inputencoding{`*encoding name*`}` ,

mit dem auch mitten im Text die Kodierung bei Bedarf umgeschaltet werden kann. (Dies kann allerdings ggf. zu Problemen führen, dazu komme ich später noch in Abschnitt 5.)

3 Einige Anmerkungen zu meinen Kodetabellen

- Die Verwendung einiger Zeichen setzt installierte $\mathcal{AMS}$ Zeichensätze [4] voraus, das sind:

  $\blacksquare$ ¥ $\ulcorner$ $\urcorner$ ® $\llcorner$ und $\lrcorner$

- Die Verwendung einiger Zeichen setzt installierte Wasy2 Zeichensätze [5, 6] voraus, dies sind:

  ¤ ¢ ‰ und ¦

- Einige mathematischen Sonderzeichen vom Atari setzen ein geladenes wasysym Paket [6] voraus, dies sind:

  ⌐ und $\oint$

- Leider sind noch nicht alle Zeichen in dieses Paket eingearbeitet, die fehlenden wurden einfach zu `\NotImplemented` definiert. Falls jemand ein Zeichen vermisst und einen Implementationsvorschlag einzubringen hat, so möge er mir bitte bitte eine Mail zukommen lassen! Ein paar Zahlen: Betroffen sind 36 Zeichen in der Datei `atari.def`, 25 in `pc850.def`, 0 in `isolatin.def`, 11 in `mac.def` und 1 in `roman8.def`.

- Zu allem Überfluß sind einige der fehlenden Zeichen gnadenlos auf „falsche“ Zeichen abgebildet worden. Was soll das? Nun ja, leider haben nicht immer alle Buchstaben in ihrer Kodierung einen passenden Partner; dieser wird aber unbedingt für korrekt funktionierende `\uppercase` und `\lowercase` Befehle benötigt, mit denen Zeichenketten in Groß- bzw. Kleinschrift ungewandelt werden.

  Hier nun eine Liste der Vermissten (und wohin sie gezwängt wurden):

Kodierung	vermisste Zeichen	verlegt nach
atari	Â Ê Ë È Ï Î Ì Ô Ò Û Ù Ÿ SS Á Ď Ó Ú	`"C0` ... `"D0`
pc850	Ÿ SS (SS = `\uppercase{ß}`)	`"B0` und `"B1`
isolatin	SS Ÿ	`"80` und `"81`
mac	SS	`"F0`
roman8	SS	`"FF`

  In späteren Versionen werde ich dieses Verhalten eventuell konfigurierbar machen.

4 Warum noch eine ISO 8859/1 Definitionsdatei?

Die ISO 8859/1 Definitionsdatei (`isolatin.def`), die diesem Paket beiliegt, besitzt einige Vorteile gegenüber derjenigen, die Teil des Inputenc-Paketes ist (`latin1.def`):

- Es sind mehr Zeichen definiert (mit Hilfe der $\mathcal{AMS}$- und Wasy2-Zeichensätze).

- Die Datei kann auch *nach* dem German-Paket (bzw. germanb aus dem Babel-Paket) eingelesen werden, ohne das man dies mit `\mdqoff` und `\mdqon` klammern müsste.

- Alle Zeichen werden in den Seitenköpfen korrekt in Großbuchstaben umgewandelt (falls der Seitenstil `headings` ausgewählt ist), wie z.B. ‘ß’ $\mapsto$ ‘SS’.

- Die Befehle `\uppercase` und `\lowercase` machen auch vor den neu zur Verfügung stehenden Zeichen nicht halt.

Natürlich gilt das oben gesagte auch für alle anderen Definitionsdateien, die diesem Paket beiliegen!

5 Probleme mit fremden Definitionsdateien

Werden die beiliegenden Definitionsdateien zusammen mit anderen verwendet, so kann es unter Umständen zu Fehlermeldungen kommen. So z.B. bei dem folgenden Beispiel:

```
\documentclass{article}
\usepackage[woanders her]{inputenc}[1994/12/10]
...
\begin{document}
...
\tableofcontents ...
\inputencoding{vom Umlaute-Paket}
...
\section{Hein Blöd fiel in den Fluß}
\end{document}
```

Nach dem zweiten Durchlauf durch LaTeX kommt es zu folgender Fehlermeldung:

```
(error.toc
! Undefined control sequence.
<argument> ...in Bl\"od fiel in den Flu\as@active
                                                 {ss}
l.1 ...n Bl\"od fiel in den Flu\as@active {ss}}{1}

?
```

Warum dies? Nun ja, einige Zeichen werden in meinen Definitionsdateien indirekt, d.h. über Macros wie `\as@active`, `\textblacksquare`, `\mathbullet` u.s.w. definiert. In diesem Falle wurde das Macro `\as@active` in die `.toc`-Datei geschrieben, weil halt ß — einer der betroffenen Zeichen – in einer Überschrift verwendet wurde. Beim nächsten Durchlauf ließt LaTeX diese `.toc`-Datei ein und gibt eine Fehlermeldung aus, da `\as@active` zu diesem Zeitpunkt noch nicht bekannt ist (es wird erst bei `\inputencoding{`*vom Umlaute-Paket*`}` dem LaTeX bekannt gemacht)!

Was nun? Nun, eine einfache Lösung wäre die Zeile

```
\usepackage[woandersher]{inputenc}[1994/12/10]    .
```

durch

```
\usepackage[irgendwas von Umlaute]{inputenc}[1994/12/10]
\inputencoding{woandersher}
```

zu ersetzen. Als *irgendwas von Umlaute* kann hierbei `atari`, `pc850`, `isolatin`, `mac` oder `roman8` eingesetzt werden, welche ist egal (denn in jeder meiner Dateien wird jedes meiner Macros definiert, um solche Probleme bei Verwendung meiner Definitionsdateien zu vermeiden).

Einige Fragen sind jedoch noch offen geblieben: Warum sind einige Zeichen über das Macro `\as@active` definiert? Weil sie ansonsten nicht in den Seitenköpfen (bei `pagestyle{headings}`) korrekt in Großbuchstaben gewandelt werden würden![1] Betroffen sind hierbei die Zeichen `\aa`, `\ae`, `\^{\i}`, `\'{\i}`, `` \`{\i} ``, `\"{\i}`, `\o`, `\ss`, `\oe`, `\dh` und `\th`.

6 Wichtiges für emTEX-Anwender

Im LATEX 2ε-Format darf keine Zeichenumwandlungstabelle enthalten sein! (Solche werden mit den Schalter `/c` eingebaut.) Stattdessen muß beim Generieren des LATEX 2ε-Formates die Option `/8` angegeben werden, z.B. so:

```
tex386 /i /8 latex.ltx
```

Ansonsten wird das Inputenc-Paket nicht funktionieren und deswegen wird man auch keine fremden Dokumente übersetzen können, die dieses Paket verwenden. Man ist also nicht in der Lage, Texte mit „fremden“ Sonderzeichen übersetzen zu können, da die Umwandlungstabelle dem Inputenc-Paket die Zeichen quasi vor der Nase wegschnappt und (falsch) umwandelt!

7 Wichtiges für CS-TEX und MultiTEX-Anwender

CS-TEX besitzt leider eine fest eingebaute Routine, die Atari-Sonderzeichen in die Cork-Kodierung umwandelt, so daß das Inputenc-Paket hier nicht funktioniert.

Was kann man dagegen machen?

1. Das Inputenc-Paket nicht benutzen!
   Mit der eingebauten Umwandlungsroutine kann man recht viele Zeichen (wie die Umlaute und das ß) auch direkt benutzen, man muß nur die Cork-kodierten Zeichensätze (`T1` Kodierung, DC-Zeichensätze) verwenden. Dies kann mit `\usepackage[T1]{fontenc}` in der Präambel des Dokumentes erreicht werden und ist sowieso aufgrund der Vorteile der DC-Zeichensätze gegenüber der CM-Zeichensätze anzuraten. Bevor man die Texte dann weitergibt, fügt man ein `\usepackage[atari]{umlaute}` für die anderen ein.

   Natürlich ist dies nur eine Nothilfe, denn schließlich ist man nicht dazu in der Lage, Dokumente mit fremden Sonderzeichen korrekt zu übersetzen!

2. CS-TEX patchen!
   Dies ist einfacher als es zunächst den Anschein haben mag:
   Man suche die Hex-Codes

   `00 01 02 03` ... `7E 7F FC DC C2 C3` ... `BE 9E`

   in den Dateien `INITEX_L.TTP` und `(TT-)TEX_L.TTP` und ersetze sie durch

[1] Dies hat Gründe, die tief in den LATEX-Quellen verborgen sind. Der Trick, wie dies erreicht wird, ist übrigens dem `german.sty` [7] entnommen.

00 01 02 03 ... 7E 7F 80 81 82 83 ... FE FF .

Diesem Paket liegt auch (in Form des uucodierten ZIP-Archivs `cs_patch.uue`) ein Programm bei, welches diese Arbeit für einen erledigt.[2]

Natürlich ist diese Lösung der ersteren vorzuziehen.

8 Andere TEX Implementationen

Falls es Probleme bei der Benutzung der beiliegenden Definitionsdateien für das Inputenc-Paket gibt (und vielleicht sogar schon eine Lösung gefunden wurde), so möge man mir bitte eine E-Mail an folgende Adresse zukommen lassen:

`Harald_Sommerfeldt@w.maus.de`

bzw.

`Harald Sommerfeldt @ W` im MausNetz.

Für Anmerkungen und Verbesserungsvorschläge bin ich ebenfalls dankbar.

Literatur

[1] Axel Sommerfeldt: *The umlaute package (v1.2)*, 1994/11/28

[2] Alan Jeffrey: *An input encoding package for LATEX 2ε (v0.09)*, 1994/12/10

[3] Helmut Kopka: *LATEX– Eine Einführung*, 4. überarbeitete und erweiterte Auflage, Addison-Wesley, Bonn, 1992

[4] *User's Guide to AMSFonts Version 2.2*, January 1995

[5] Roland Waldi: *The Symbol font wasy*, Version 2.0 - September 1992

[6] Axel Kielhorn: *The wasy symbol fonts for use with LATEX 2ε*, 94/05/28

[7] Bernd Raichle: *Style-Option "german" für plain-TEX, LATEX 2.09, LATEX 2ε und anderen auf diesen basierenden Formate*, 20. Januar 1995

[2] An dieser Stelle ein herzliches Dankeschön an Markus Kohm für das Programm!