



Magazin-/Bestellzetteldruck an der UB Bayreuth

SSV/ASP-Treffen 28.06.2012

Dr. Andreas Weber

IT-Servicezentrum

Universität Bayreuth

Andreas.Weber@uni-bayreuth.de



Das Problem...



Das Problem...

Anforderungen an den Magazinzetteldruck nach der Umstellung auf Unicode (SunRise V4):

1. Die Ausgabe von Unicode-Zeichen auch auf den Bestell-/Magazinzetteln gewünscht.
2. Barcodes für AFL etc. und Hervorhebungen in der Fußzeile sollen weiterhin auf die Zettel gedruckt werden.
3. Der Druckvorgang soll nicht wesentlich langsamer werden.

Lösung mit einem „Skript“ ...

Statt des direkten Drucks der Zettel werden diese mit einem Programm „behandelt“.

Die Aufgaben des Skripts:

1. Parsen und Verändern des Inhalts des Zettels
2. *Zuordnung der Drucker aus Signaturanfang (UBT)*
3. - Einfügen von Barcodes
- Einfügen einer Fußzeile mit Hervorhebungen
4. **Darstellung der UTF-8 Zeichen**
5. **Optimierung der Geschwindigkeit**



Skript statt Druckkommando...

	Gerätebezeichnung	Druckersequenz
Ausleihe		
Systemparameter		Bitte geben Sie als Platzhalter für den Druck durch den Namenseintrag der Magazindruck
Benutzerparameter		
Zweigstellenverwaltung		
Tabellendefinitionen		
Signatur-/Druckerverwaltung		
Druckertabelle	4813	/opt/lib/sisis/local/bin/ZettelDruck_dummy <l
Druckersequenzen	9014	/opt/lib/sisis/local/bin/ZettelDruck_dummy <l
Magazindruckertabelle	high	/opt/lib/sisis/local/bin/ZettelDruck_dummy <l
	tech	/opt/lib/sisis/local/bin/ZettelDruck_dummy <l

`/opt/bin/startdruck`

```
PATH=$PATH:/opt/lib/sisis/local/bin
export PATH
LPR1="ZettelDruck_dummy"
LPR2=""
export LPR1 LPR2
```

- Ausgangspunkt:
 - Ip.xxx Dateien aus dem spool-Verzeichnis des AV-Servers
 - Ip.xxx sind reine Textdateien **mit UTF-8 Zeichen**
- Parsen, Verändern, Druckerzuordnung:
 - Lesen der Datei und Ermitteln von Informationen
 - Welche Art von Zettel liegt vor
 - Besondere Inhalte (TEMP-Nummer, ...)
 - Verändern des Inhaltes
 - Zeilen weglassen
 - Abkürzungen
 - *Ermittlung des Druckers anhand der Signatur (UBT)*

UTF-8 Zeichen drucken...

Drucker haben meist keinen UTF-8 Zeichensatz:

→ der Font muss dem Drucker mit geliefert werden

→ Zeichen werden als Grafiken erzeugt

Beispiel: **text2ps** von Cups

→ wandelt ein Textfile in ein (großes) PS-File um

→ benötigt Fonts, z.B. Freefonts und eine PPD Datei

```
setenv CHARSET utf-8
setenv PPD /usr/local/sisis-pap/cups/etc/cups/ppd/btu0pd.ppd
cat /var/spool/sisis/avserver/batch/tmp/lp.3.10394.1 \
|/usr/local/sisis-pap/cups/lib/cups/filter/texttops 1 1 1 1 1
```

Für die Umwandlung gibt es weitere Lösungen, z.B.

[Pango](#)



Barcode als Postscript hinzufügen:

<http://code.google.com/p/postscriptbarcode/>



The screenshot shows the Google Code project page for 'postscriptbarcode'. The page title is 'postscriptbarcode' with the subtitle 'Barcode Writer in Pure PostScript'. There is a search bar and navigation links for 'Project Home', 'Downloads', 'Wiki', 'Issues', and 'Source'. The 'Summary' tab is selected, showing project information, labels, members, and featured downloads. The main content area is titled 'Barcode Writer in Pure PostScript' and contains a description of the project, a list of supported barcode formats, and a code snippet for adding a barcode to a PostScript document.

postscriptbarcode
Barcode Writer in Pure PostScript

Project Home Downloads Wiki Issues Source

Summary Updates People

Project Information

★ Starred by 55 users
[Activity](#) ■ Medium
[Project feeds](#)

Code license
[MIT License](#)

Labels
Barcode, PDF417, PostScript, EAN, UPC, ISBN, DataBar, Generator, ITF, DataMatrix, Aztec, QRCode, Codabar, MicroPDF417, GS1

Members
[terry.burton](#)

Featured

Downloads
[barcode.ps](#)
[postscriptbarcode.tgz](#)

Barcode Writer in Pure PostScript

Barcode Writer in Pure Postscript implements the printing of all major barcode formats entirely within PostScript, so that the process of converting the input string into the printed output is performed by the printer itself. This avoids the need to re-implement the barcode generation process whenever your language needs change.

It makes including any barcode within a PostScript document as simple as inserting the following directive:

```
0 0 moveto (978-1-86074-271) (includetext) /isbn /uk.
```

Features

The project supports all major barcode formats including:

[EAN-13](#), [EAN-8](#), [UPC-A](#), [UPC-E](#), [ISBN](#), [ISMN](#), [ISSN](#), [EAN-5](#) & [Expanded Stacked](#), [GS1 DataBar Stacked Omnidirectional](#), [GS1 DataBar 1D](#), [Code 128](#), [GS1-128 \(EAN-128\)](#), [SSCC-18 \(EAN-18\)](#), [Code 39](#), [Code 39 Extended](#), [Code 93](#), [Code 93 Extended](#), [Code 2 of 5](#), [ITF-14 \(UPC SCS\)](#), [Code 2 of 5 \(Industrial, IATA\)](#), [Code 2 of 5 \(USPS\)](#), [Code 2 of 5 \(Code 2 of 5\)](#), [ITF-14 \(UPC SCS\)](#), [Code 2 of 5 \(Code 2 of 5\)](#), [BC412](#), [Codabar \(NW-7\)](#), [Pharmacode](#) (including [Pharmacode](#)), [Channel Code](#), [PosiCode](#), [PDF417](#), [MicroPDF417](#), [Data Matrix](#), [GS1 DataMatrix](#), [QR Code](#) (including [Micro QR Code](#)), [MaxiCode](#), [Aztec Code](#), [USPS Intelligent Mail](#), [USPS POSTNET](#), [USPS PLANET](#), [Royal Mail \(RM4SCC\)](#), [KIX](#) (Dutch Postal), [JapanPost](#), [AusPost](#), [Deutsche Post Identcode](#), [Deutsche Post Leitcode](#), [USPS FIM symbols](#), [DAFT](#), [Raw](#), [Fluttermarken](#), [HIBC barcodes](#) (Code 39, Code 128, Data Matrix, PDF417, MicroPDF417, QR Code, Codablock F)

```
$zus_text .= sprintf(" %s %s moveto (%s) (includetext width=2 height=0.2) /code39 /aw_ps findresource exec\n", $xpos, $ypos, $AFL );
```



Fußzeile als PostScript einfügen

```
# -- Inhalt der letzten Zeile
my $last_line = sprintf(
    "%-20s  -%11s-  %15s%30s",
    substr( decode( "utf-8", $NAME ), 0, 20 ),
    $BENNR,
    substr( $AUSGABE, 3, 15 ),
    substr( $SIGNATUR, 0, 30 )
);
if ( $DEBUG > 0 ) {
    print STDERR " -> Fusszeile: $last_line\n";
}

$zus_text .= sprintf("/FussFont findfont 12 scalefont setfont\n");
$zus_text .= sprintf( "%s %s moveto (%s) show\n", 12, 430, $last_line );

$zus_text .=
    sprintf("grestore\nshowpage\n");
```

```
printf("/FussFont findfont 12 scalefont setfont\n");
printf( "%s %s moveto (%s) show\n", 12, 430, $last_line );
```



Optimierung (1)

Die entstehenden Dateien sind sehr groß...

→ Umwandeln mit Ghostscript

→ Verschiedene Endformate wählbar

```
#!/usr/bin/ksh

#$1 Input Postscript
#$2 Header.ps
#$3 Outputfile
#$4 Ausgabeformat

case $4 in
  pcl) FORMAT=pcl3
        ;;
  ps)  FORMAT=pswrite
        ;;
  pdf) FORMAT=pdfwrite
        ;;
  *)   FORMAT=pswrite
        ;;
esac

cat $2 $1 | /opt/csw/bin/gs -q -sDEVICE=$FORMAT -sOutputFile=$3 -_
return $?
```



Jeder Zettel wird einzeln gedruckt, was bei vielen Zetteln, wegen der Übermittlung der Header, ein Zeitproblem ergibt.

→ Erzeugen eines mehrseitigen Ausdrucks für jeden Drucker

a. Verwendung der Sammeldateien `lp.bs.<pid>`

→ Drucker aus Signatur ermittelt wird

b. Erzeugen einer Sammeldatei im „Dummy-Druck“, wobei der Drucker pro Zettel in der Sammeldatei vermerkt wird.

/opt/bin/startdruck

```
PATH=$PATH:/opt/lib/sisis/local/bin
export PATH
LPR1="ZettelDruck_dummy"
LPR2=""
export LPR1 LPR2
```

```
PATH=$PATH:/opt/lib/sisis/local/bin
export PATH
LPR1="ZettelDruck_dummy"
LPR2=""
export LPR1 LPR2
```

```
$BATCHROOT/bin/prg/BSDRUCK $@
rc=$?
```

```
# ermittle die letzte Sammeldatei lp.bs.*
WDIR=/var/spool/sisis/avserver/batch/tmp
# letzte bearbeitet Datei
OLD=`cat $WDIR/old_lp_file`
LPDATEI=`ls -tr $WDIR/lp.bs* | tail -1`
# ist die neue Datei schon da?
count=0
# warte, bis sich da was aendert
while [ "x$OLD" == "x$LPDATEI" ]
do
  ((count=count+1))
  if [ $count -gt 10 ]
  then
    rm -f $BATCHSPOOL/bsdruck.lock
    exit 1
  fi
  LPDATEI=`ls -tr $WDIR/lp.bs* | tail -1`
  sleep 5
done
```

```
# ermittle die letzte Sammeldatei lp.bs.*
WDIR=/var/spool/sisis/avserver/batch/tmp
```

```
#Drucke die Sammeldatei
ZettelDruck bib $LPDATEI
echo $LPDATEI > $WDIR/old_lp_file

rm -f $BATCHSPOOL/bsdruck.lock
exit $rc
```

```
#Drucke die Sammeldatei
ZettelDruck bib $LPDATEI
```



Ergebnis

***** UB Bayreuth *

M A G A Z I N - B E S T E L L U N G

Vom: 10.02.2012 Terminal: opacweb Zweigst: 00 9:24:31 Uhr
Name Zweigstelle : Campus
Benutzernummer: 01760027110
Name : Weber, Andreas Thomas

Verfasser : Şālihānī, Anṭūn
Titel : Alf laila wa-laila/1
Mediennummer : 017056582236
AFL-Nummer : @201204864
Signatur : 00/EN 3040.956-1
Ausgabeort : 04 Teilbibliothek NW II

Verfasser : Şālihānī, Anṭūn
Titel : Alf laila wa-laila/1 th *



Weber, Andreas Thoma -01760027110-

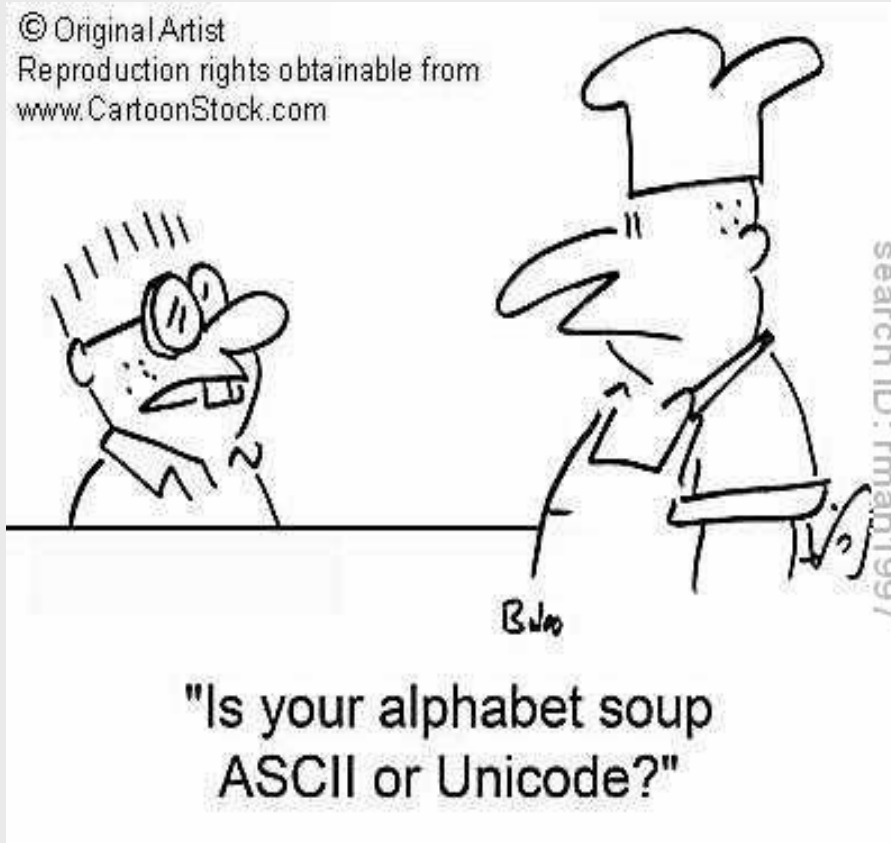
TB NW II

00/EN 3040.956-1



Vielen Dank für Ihre
Aufmerksamkeit!

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



Fragen?

Jetzt

oder später an

Dr. Andreas Weber
IT Servicezentrum
Universität Bayreuth
<http://www.rz.uni-bayreuth.de>
andreas.weber@uni-bayreuth.de

