
ASCII.sty (version 1)
A new style-option and encoded font with
IBM graphics control characters for use
with T_EX & L^AT_EX.

R. Ramasubramanian
 R.W.D. Nickalls
 M.A. Reed

Abstract

This article describes a new font and style option (ASCII.STY) which includes the IBM graphics characters commonly used by PCs to represent the control codes (ASCII 1–31). In particular, this article does three things: (a) it announces and describes a pk rendering of the Type 1 Courier font donated to the X Consortium by IBM; (b) a L^AT_EX style option is presented which uses this font; (c) the method used to create the font is described in detail.

1 Introduction

ASCII.STY¹ and the related font files shown in Table 1 which constitute the package ‘ASCII’, were developed while we were writing a book on *serial interfacing*². We were using L^AT_EX and wanted to include the usual table of ASCII control codes and their equivalent IBM graphics characters as used by PCs. However, in spite of searching all the available archives, we were unable to find a T_EX font containing *all* the graphics characters of the 7-bit ASCII code (e.g. © ª ¯ ¯ ¯ ¯ ¯ ¯ ¯ etc.).

One of us (Mark Reed) noticed that the same problem was raised in a recent issue of UKT_EX Digest³, in which it was suggested that a possible solution might be to derive the characters from the public domain PostScript IBM COURIER font donated by IBM to the X Consortium.

We therefore downloaded the IBM COURIER font, which consists of two files COUR.AFM and COUR.PFA, and duly got to work extracting the characters and generating the necessary .TFM and bit-mapped .PK files.

In practice, however, it was not at all easy owing to a paucity of documentation on just exactly how to do this. This article therefore details what

¹ The complete package of files (ASCII) described in this article (see Table 1) can be found in the CTAN archive (/FONTS/ASCII).

² Nickalls RWD and Ramasubramanian R (1994). *Interfacing the IBM PC to medical equipment: the art of serial communication*. Cambridge University Press (in press).

³ See note by Sebastian Rahtz on ‘AFM files for IBM Courier’ in UKT_EX Digest, Friday 18 Dec 1992, Vol 92: Issue 47.

we did and how we did it, and ASCII.STY is the result.

In order to avoid any confusion, and to clearly distinguish the new derivative font from the original IBM COURIER font, this document will refer to the new derivative font as ASCII-10.

We have restricted our efforts to the first 128 characters of the PC character set because the term ASCII, as used by the original ANSI standard⁴, refers only to the first 128 characters.

Those who are interested in unmasking the characters 128–255 in the PC character set may wish to examine some of the other .AFM files held in the CTAN archive (e.g. CRB-PC8.AFM; CRI-PC8.AFM).

2 The ASCII font

Table 2 shows the 10-point ASCII-10 font which we derived from the IBM COURIER font. Different sizes can be generated using the T_EX magstep procedure.

Each element in the font table can be accessed in the usual way by using the `\char` command. For example, the style option ASCII.STY allocates the command `\CR` for the Carriage Return character as follows.

1. The new font is defined as `\ascii` using `\newfont{\ascii}{ascii-10}`
2. The new command `\CR` is defined using `\newcommand{\CR}{\char'015}`.

Now when the command `{\ascii\CR}` is used in the text it produces the symbol ª. The brackets are needed to keep the `\ascii` font local.

3 Installing the font files

EmT_EX implementation

In order to use ASCII.STY the various font files (.PK and .TFM files) need to be placed in the correct directories, as specified by EmT_EX.

ASCII-10.TFM file: This file goes into the usual `\EMTEX\TFM` directory.

ASCII.FLI library: This library of all ASCII.PK files goes into the usual `\TEXFONTS` directory.

Non-EmT_EX implementations

For those not using EmT_EX the separate ASCII.PK files are also provided. Although all the .PK files initially have names which indicate the dpi size (e.g. ASCII300.PK), once correctly located in their respective subdirectories, they must all then be renamed so that they all have the same name ASCII-10.PK.

⁴ American National Standard X3.4 (1986). Available from the American National Standards Institute, 1430 Broadway, New York, 10018.

File name	Size*	Description
ASCII.STY	5805	The ASCII style option.
ASCIISTY.TEX	27437	Source file for ASCII package documentation.
ASCIISTY.DVI	38796	ASCII package documentation .DVI file.
ASCII-10.TFM	772	T _E X Font Metric file for ASCII-10 font.
ASCII.ENC	2459	The ASCII-10 PostScript encoding vector used by AFM2TFM and PS2PK to rearrange the font.
ASCII.FLI	55121	Font library containing all the ASCII.PK fonts.
ASCH300.PK	5140	10-point ASCII-10 font file.
ASCH329.PK	5588	11-point ASCII-10 font file.
ASCH360.PK	6016	12-point ASCII-10 font file.
ASCH432.PK	7068	14-point ASCII-10 font file.
ASCH518.PK	8596	17-point ASCII-10 font file.
ASCH622.PK	10188	21-point ASCII-10 font file.
ASCH746.PK	12336	25-point ASCII-10 font file.
UKTEXDIG.DOC	1783	A short note by Sebastian Rahtz.
COUR.AFM	29886	Adobe Font Metric file for IBM COURIER font.
COUR.PFA	128018	PostScript Font ASCII file for IBM COURIER font.
COPYRIGH.DOC	523	The copyright notice by IBM.
A4.STY	11070	Style option needed for typesetting ASCIISTY.TEX.
README.DOC	2987	Information for unpacking ASCII.STY.

Table 1: List of all files contained in the ASCII package (CTAN archive /fonts directory).

* The size given here relates to the file size in bytes on IBM compatible PCs. The point size of the .PK files relates to devices having a 300 dpi resolution.

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x		⊙	⊛	♥	♦	♣	♠	•	"0x
'01x	■	○	⊗	♂	♀	♯	♭	⊛	
'02x	▶	◀	‡	!!	¶	§	—	‡	"1x
'03x	↑	↓	→	←	L	↔	▲	▼	
'04x		!	"	#	\$	%	&	'	"2x
'05x	()	*	+	,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	<	=	>	?	
'10x	@	A	B	C	D	E	F	G	"4x
'11x	H	I	J	K	L	M	N	O	
'12x	P	Q	R	S	T	U	V	W	"5x
'13x	X	Y	Z	[\]	^	_	
'14x	'	a	b	c	d	e	f	g	"6x
'15x	h	i	j	k	l	m	n	o	
'16x	p	q	r	s	t	u	v	w	"7x
'17x	x	y	z	{		}	~	∅	
	"8	"9	"A	"B	"C	"D	"E	"F	

Table 2: The ASCII-10 font.

3.1 Adjusting the configuration file (LJ.CNF)

In order that EmTeX can find all fonts, the path for the fonts in the LJ.CNF file (in the \EMTEX directory) may need to be altered to accommodate the directory structure of \TEXFONTS.

The authors use the following library line (/pl) in the LJ.CNF file.

```
/pl=%DVIDRVFONTS%;lj_0;lj_h;lj_1;lj_2;
lj_3;lj_4;lj_5a;lj_5b;lj_6;ascii;
```

4 Using ASCII.STY

Place ASCII.STY in the \TEXINPUT directory, and then use the style option [ASCII] in the \documentstyle command in the usual way. ASCII.STY includes seven magnifications of the ASCII-10 font as follows: magstep 0, half, 1, 2, 3, 4, 5, which are equivalent to point sizes 10, 11, 12, 14, 17, 21, 25 respectively. Note that the relationship between point size and magstep is such that the point size corresponding to magstep n is 10×1.2^n , i.e. magstep 0 is equal to 10-point size.

Using ASCII.STY an ASCII character is most easily printed using the \ascii command followed by the character in the usual way. For example, {\ascii 5} will print the number 5 from the ASCII-10 font table. The ASCII.STY style option will automatically select the font size currently being used in the document (e.g. 10, 11, 12 point). For convenience all the control characters except the four DC₁₋₄ have been allocated their usual code name (in *uppercase* letters); thus the carriage return symbol (↵) is called \CR and would be printed using {\ascii\CR}. The four Device Control characters DC1, DC2, DC3, DC4 have been allocated the commands \DCa, \DCb, \DCc, \DCd, since TeX does not allow commands to contain integers.

Two other characters in the ASCII-10 font table which are not usually available for TeX users are

- Δ (\DEL) which is printed using {\ascii\DEL}.
- | (\splitvert) which is printed using {\ascii\splitvert}. This name was chosen to be in keeping with Knuth's \vert which is |.

The standard codes and the coding used by ASCII.STY for the control characters are shown in Table 3.

4.1 Using other sizes

When an ASCII character having a different point size to the one being used for the document is required, then a different command without integers has to be used.

In view of this restriction ASCII.STY uses Roman numerals to indicate both point size (e.g.

\asciixii for 12-point size) and magstep number (e.g. \asciimsV for magstep 5), as shown in Table 4. For example, a 21-point heart shape ♥ can be printed using either of the commands {\asciixxi\ETX} or {\asciimsiv\ETX}.

Examples of the characters and sizes given by the commands are shown below.

The 10 point font: {\asciimsZ\S0} and \asciix\S0} give	♫
The 11 point font: {\asciimsH\S0} and \asciixi\S0} give	♫
The 12 point font: {\asciimsI\S0} and \asciixii\S0} give	♫
The 14 point font: {\asciimsII\S0} and \asciixiv\S0} give	♫
The 17 point font: {\asciimsIII\S0} and \asciixvii\S0} give	♫
The 21 point font: {\asciimsIV\S0} and \asciixxi\S0} give	♫
The 25 point font: {\asciimsV\S0} and \asciixxv\S0} give	♫

5 Generating the ASCII font.

As indicated earlier, the IBM COURIER font files (COUR.AFM and COUR.PFA) contained all the graphics characters needed for typesetting a 7-bit ASCII table. In order to make use of this font in a non-PostScript TeX or LaTeX environment, we needed to generate the following:

- A .TFM file from the corresponding PostScript .AFM file.
- An MS-DOS binary .PFB font file from the corresponding .PFA font file (a Unix text file).
- A bit-mapped .PK font file from the corresponding .PFB file.

The first step requires a program called AFM2TFM; the second step requires a program called T1BINARY; and the third step requires the program PS2PK. Fortunately, all the programs required for performing these conversions on a PC are readily available from ftp.urc.tue.nl [131.155.2.79] in their /TEX/EMTEX directory. The various steps are shown in Figure 1.

5.1 Generating the .TFM file

Normally this would be straightforward since the AFM2TFM program reads a PostScript font metric information file (i.e. an .AFM file) and produces a .TFM file.

In practice, it was not easy to establish which characters were in the COUR.AFM file since there was no documentation. For example, we could only discover by trial and error that the line

ASCII control codes						
<i>Dec</i>	<i>Code</i>	<i>Hex</i>		<i>Name</i>	<i>IBM</i>	<i>ASCII.STY</i>
0	NUL	00h	~@	Null	—	—
1	SOH	01h	~A	Start of heading	☺	\SOH
2	STX	02h	~B	Start of text	☹	\STX
3	ETX	03h	~C	End of text	♥	\ETX
4	EOT	04h	~D	End of transmission	♦	\EOT
5	ENQ	05h	~E	Enquiry	♣	\ENQ
6	ACK	06h	~F	Acknowledge	♠	\ACK
7	BEL	07h	~G	Bell	•	\BEL
8	BS	08h	~H	Back space	▣	\BS
9	HT	09h	~I	Horizontal tab	○	\HT
10	LF	0Ah	~J	Linefeed	◻	\LF
11	VT	0Bh	~K	Vertical tab	♂	\VT
12	FF	0Ch	~L	Formfeed	♀	\FF
13	CR	0Dh	~M	Carriage return	♪	\CR
14	SO	0Eh	~N	Shift out	♫	\SO
15	SI	0Fh	~O	Shift in	♬	\SI
16	DLE	10h	~P	Data link escape	▶	\DLE
17	DC1	11h	~Q	Device control 1 (XON)	◀	\DCa
18	DC2	12h	~R	Device control 2	‡	\DCb
19	DC3	13h	~S	Device control 3 (XOFF)	!!	\DCc
20	DC4	14h	~T	Device control 4	¶	\DCd
21	NAK	15h	~U	Negative acknowledge	§	\NAK
22	SYN	16h	~V	Synchronous idle	—	\SYN
23	ETB	17h	~W	End transmission block	‡	\ETB
24	CAN	18h	~X	Cancel	↑	\CAN
25	EM	19h	~Y	End of medium	↓	\EM
26	SUB	1Ah	~Z	Substitute (EOF)	→	\SUB
27	ESC	1Bh	~[Escape	←	\ESC
28	FS	1Ch	~\	File separator	⌞	\FS
29	GS	1Dh	~]	Group separator	↔	\GS
30	RS	1Eh	~^	Record separator	▲	\RS
31	US	1Fh	~_	Unit separator	▼	\US
124	—	7Ch	—	—	⋮	\splitvert
127	DEL	7Fh	—	Delete	⊠	\DEL

Table 3: ASCII table of control characters.

Using point sizes		Using magstep Nos.	
10	\asciix	Zero (msZ)	\asciimsZ
11	\asciixI	Half (msH)	\asciimsH
12	\asciixII	1 (msI)	\asciimsI
14	\asciixIV	2 (msII)	\asciimsII
17	\asciixVII	3 (msIII)	\asciimsIII
21	\asciixXI	4 (msIV)	\asciimsIV
25	\asciixXV	5 (msV)	\asciimsV

Table 4: Font size commands for point and magstep, as used by ASCII.STY.

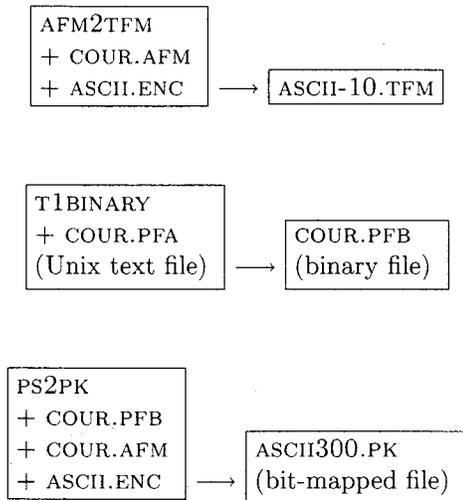


Figure 1: Sequence of file changes to generate ASCII-10.TFM and ASCII300.PK.

```

C -1 ; WX 600 ; N SV040000 ;
B 102 96 498 467 ;

```

referred to the ASCII character \blacktriangledown . Note that the C -1 code indicates that the character is *not* 'visible'.

Once we had established the relative positions of the characters in the .AFM file, it was fairly easy to generate an 'encoding vector' file (ASCII.ENC). The following is an extract from the ASCII.ENC file:

```

/asciiEncoding [
  /spacnumeric /SS000000 /SS010000
  /heart /diamond /club
  /spade /bullet /SM570001
  :
  :
  /.notdef /.notdef /.notdef /.notdef
] def

```

The ASCII.ENC file contains 128 character names each preceded by /. The order of the names in this file specifies the order of the characters in the \TeX font. Any character position which is not defined (i.e. those character positions with a value > 127) are designated /.notdef.

Once we had generated a complete ASCII.ENC file, the ASCII-10.TFM file was generated using the following command.

```

afm2tfm cour.afm -T ascii.enc
ascii-10.tfm

```

Newer versions of AFM2TFM will automatically print out a line suitable for inclusion in PSFONTS.MAP for use with DVIPS. **Ignore** the automatically generated line and see the section below on using DVIPS for more information.

For further information on AFM2TFM, the reader should refer to the documentation which accompanies this program. Readers who wish to know more about the codes found in .AFM files should consult the PostScript Language Reference Manual [1st edition] published by the Adobe Corporation.

5.2 Conversion of .PFA to .PFB format

When PS2PK is run on PCs with MS-DOS, the program will only read .PFB files and not .PFA files. In view of this, it was necessary to first convert the .PFA font file (which is a Unix text file), into an MS-DOS binary (.PFB) font file using the program T1BINARY, using the following command:

```
t1binary cour.pfa cour.pfb
```

5.3 Generating the .PK bit-mapped file

The .PFB file generated in the previous step, can now be used by PS2PK to produce a .PK font file. This is done with the following command (note that PS2PK requires that COUR.AFM is also in the same subdirectory):

```
ps2pk -e ascii.enc cour.pfb ascii300.pk
```

This produces a 10-point ASCII300.PK file (i.e. ASCII-10) for an output device with a resolution of 300dpi. For further information regarding the command line switches and the use of PostScript fonts with \TeX and \LaTeX , please read the detailed documentation which comes with the PS2PK package.

6 Magnifying the ASCII-10 font

As indicated earlier, the steps described above will produce the font at its optimal 10-point design size i.e. ASCII-10 (since it was derived from the 10-point IBM COURIER font) and for an output device with a resolution of 300dpi.

In order to produce an ASCII table with larger size graphics characters in it, additional .PK files must be generated (e.g. ASCII360.PK) which contain magnified versions of these characters. This is done using PS2PK with the -X flag as follows.

```
ps2pk -e ascii.enc -X360 cour.pfb
ascii360.pk
```

Using -X360 will produce a 12-point font by magnifying a 10-point design size font. By varying the -X value different magnified sizes can be obtained.

When all the necessary .PK files have been produced, they can be gathered into a single compact font library (ASCII-10.FLI in this case) using a program called FONTLIB. This program and its documentation may be found in the standard Em \TeX package in the CTAN archive.

7 Using ASCII-10 with Tomas Rokicki's DVIPS

Using any PostScript font is extremely easy with DVIPS, and ASCII-10 is no exception. Simply copy the COUR.AFM, COUR.PFB and ASCII.ENC files to DVIPS's header directory. Then add the following line to the end of PSFONTS.MAP.

```
ascii-10 Courier <cour.pfb
" asciiEncoding ReEncodeFont "
<ascii.enc
```

Don't use the line generated by AFM2TFM as it will not work!

Once DVIPS has been configured in this way, it will automatically use the PostScript outline instead of using a bitmap. This gives much better output quality and is resolution independent.

- ◊ R. Ramasubramanian *MBBS, FRCA*.
Department of Anaesthesia,
University Hospital,
Queen's Medical Centre,
Nottingham, NG7 2UH,
England, UK.
100016.3245@CompuServe.com
- ◊ R.W.D. Nickalls *BSc, PhD, MBBS, FRCA*.
Senior Lecturer and Consultant,
Department of Anaesthesia,
University Hospital,
Queen's Medical Centre,
Nottingham, NG7 2UH,
England, UK.
dick.nickalls@nottingham.ac.uk
100115.1010@CompuServe.com
- ◊ M.A. Reed
Cambridge University Press,
Technical Applications Group,
Printing Division,
University Printing House,
Shaftesbury Road,
Cambridge, CB2 2BS
England, UK.
mreed@cup.cam.ac.uk

Book Reviews

Book review: *The L^AT_EX Companion*

David M. Jones and David E. Wald

*In darkness languishes the precious stone.
When will its excellence enchant the world?*

– Barry Hughart

Michel Goossens, Frank Mittelbach and Alexander Samarin, *The L^AT_EX Companion*. Addison-Wesley, 1993. xxx + 492 pp. + index (36 pp.) ISBN 0-201-54199-8.

Without question, this book is absolutely necessary for all who use, support, modify, or rue the day that they first heard of L^AT_EX. You should all run right out and get a copy. We'll wait.

All set? Good. Now we'll tell you what you've just acquired.

In recent years there has been an explosion of books about T_EX and L^AT_EX, and the majority of L^AT_EX users have probably gone from gnashing their teeth over the lack of supplemental documentation to gnashing their teeth over which of the numerous books on the market are the best investment for their time and money. Let there be no confusion about the *L^AT_EX Companion*: the *Companion* belongs next to the *T_EXbook* and the L^AT_EX manual as a book that every L^AT_EX user should have available.

Since its introduction in the early 80's, L^AT_EX has been the center of a flurry of grassroots development, with contributors world-wide producing a variety of style files and support packages, many of which have become basic necessities to any well-dressed L^AT_EX installation. In recent years, this development has reached a minor crisis point, with several slightly incompatible variants of L^AT_EX coming into common use.

As anyone who has been following the recent development of L^AT_EX knows, the L^AT_EX3 team, spearheaded by Frank Mittelbach, Chris Rowley, and Rainer Schöpf, has taken over the development of L^AT_EX, and is in the process of producing a completely rewritten L^AT_EX3 [1-3]. Recently, the L^AT_EX3 team, in an attempt to head off the aforementioned crisis, announced the imminent release of a single "state of the art" version of L^AT_EX, to be designated L^AT_EX2_ε.