EAN Barcodes by TEX an Metafont

Peter Willadt

1998-01-24

Abstract

This article documents the use of the font `wlean.mf`. With the font, there also comes an auxiliary Perl file for preprocessing TeX source. Its use is also documented here.

1 Legal Restrictions and Introduction

All the files in this package are subject to the LATEX Project Public License. Also I got a special wish: If you find this package useful, support TUG or your local TEX user group.

The complete contents of this package is described in the file `README`. That file also contains some information about the use of the other barcode fonts that are contained. For more information about these other files you may want to read an article that has appeared in the december, 1997 issue of *TUGboat*. Please see also the remarks in the section *Address* later in this file.

2 About EAN

In stores, UPC and EAN codes are widely used for automatic identification, pricing etc. EAN consist of either eight or thirteen digits. Twelve digit UPC codes are like thirteen digit EAN with the first digit equal to zero. EAN specifications do not only require bars, also the encoded number has to be written in plain text, in case a reader is defective or the code is too hard to read. EAN is a high-density code, and so it is highly vulnerable.

2.1 Coding

The last digit of an EAN is a weighted mod 10 checksum. Digits are alternatingly multiplied by 1 or 3. The so calculated sum over all digits has to be divisible by ten without any remainder.

There are three different EAN character sets labeled A, B, and C. Eight digit EAN codes use character sets A and C, EAN codes with thirteen digits use all three character sets—see below.

EAN with eight digits consists of:
a sidebar
the first four digits (coded in character set A),
the middle separator,
the other four digits (coded in character set C)
another sidebar.

The first half of an EAN code with thirteen digits is coded in the character sets A and B, the second half in character set C. The coding of the very first digit is hidden in the varying use of the character sets A and B. A C programmer might use the following table and algorithm to decide which character set to use for digits 2–7:

```
static UBYTE abtab[10][6]={
  {0,0,0,0,0,0}, /* 0 */
  {0,0,1,0,1,1}, /* 1 */
  {0,0,1,1,0,1}, /* 2 */
  {0,0,1,1,1,0}, /* 3 */
  {0,1,0,0,1,1}, /* 4 */
  {0,1,1,0,0,1}, /* 5 */
  {0,1,1,1,0,0}, /* 6 */
  {0,1,0,1,0,1}, /* 7 */
  {0,1,0,1,1,0}, /* 8 */
  {0,1,1,0,1,0}  /* 9 */
};

char eancode[18];
char eansource[14]="4025700001030";

eancode[0]=eansource[0];
eancode[1]=' ';
eancode[2]='+';
for(i=1;i<7;i++)
  eancode[i+2]='A'+eansource[i]-'0'
               +abtab[eansource[0]-'0'][i-1]*('a'-'A');
/* then the middle separator, digits 7--13,
 * and the final + sign */
```

A zero means to use character set A and a one means to use character set B for the respective digit. The printed code of an EAN 13 consist of the following elements, from left to right:

The first digit in human-readable form
an EAN sidebar
six digits in character sets A or B
the EAN middle separator
six digits in character set C
another EAN sidebar

Magazines or codes with pricing have a so called extension following the main code with some fixed distance. This extension consists of one sidebar and two or five digits. As I have no full EAN documentation at hand at the time of this writing, I am sorry that I am not able to tell you more about this.

The EAN digits themselves obey to the following rules: Each digit takes seven units of space. Some of the seven elements are white, others are black. Digits from character set A always are white at the left edge and black at the right, and they always have an odd numer of black elements. Digits from character set B are quit similiar, but they have an even number of black elements. Digits from Character set C always start with a black element and have an even number of black elements. The rightmost element in character set C is always white. The sidebars are three elements wide, the middle separator takes five elements.

2.2 What EAN numbers may I use?

For inhouse use, you may use any 13-digit EAN that starts with a 2. If you want to have your products sold elsewhere, you have to buy a set of EAN numbers from the organisation in your country that holds these numbers. For germany, this organisation is the *Zentrale für Coorganisation* in Cologne. Almost any country has a similiar organisation.

The first digit or sometimes the first two digits code the country of origin, the next five to six digits code the manufacturer, the eigth to twelfth digits are for free use by the manufacturer. The thirteenth digit is, as explained above, a checksum. EAN do not contain any qualifiers, so if you get an EAN from somewhere, you may find out about the country of origin and about the manufacturer of the product, but if you want to know more, you have to contact the manufacturer.

3 Using `wlean.mf`

`wlean.mf` is rather raw. It contains all three EAN character sets within one single font, but at different places. The character sets A, B, C, and the digits are featured through the following characters:

0 to 9 yield the digits from 0 to 9
A to J yield the codes from character set A
a to j yield the codes from character set B
K to T yield the codes from character set C
+ makes the left and right sidebar and
- makes the middle sign

So, to code the number `2099993098253`, you have to write `{\eanfont2␣+AJjjJd-KTSMPN+}`. The space is necessary to separate the leading 2 from the barcode.

`wlean.mf` does not use true OCR digits, as it should. As the digits will not be used for OCR, I do not consider this as a serious restriction. If you really need

OCR digits, there is an OCR font on CTAN. And in *TUGboat*, there has been a publication about EAN, where TEX draws the bars and the OCR font prints the digits, see [1].

`wlean.mf` uses the normal EAN dimensions. If you would like lower bars—in contradiction to the EAN rules—you have to edit the source. The rules also make recommendations about the scaling. To be fully compatible, this font may be scaled 0.8, 0.9, 1, 1.2, 1.4, 1.5, 1.7, 1.85, or 2 times the original size. With a 300 dpi printer, I do not recommend using sizes< 1.0.

3.1 Installation

The installation itself is pretty mundane, like with any plain font. Just copy `wlean.mf` to a location where Metafont can find it. Then invoke Metafont to create a TFM file. Move this TFM file where TEX can find it. Type in the example at the end of this file and run it through TEX. Then call Metafont again to produce a font suitable for your printer or previewer and move this font to an appropriate location. You may also want to edit `codean.pl` to run on your shell. For this purpose you have to read your system's documentation or the documentation that comes with Perl.

3.2 Making readable output

Don't make EAN too small. With a 300 dpi printer, you should not use this font with magnification< 1; `scaled 1200` will be okay. If you want to do mass production, go to somebody with a barcode reader and check your output, *before* you loose money. You also should consider changes in the blackness that may be caused by production printing devices. And, of course, you should only use colours that can be used with barcode reading devices. Especially, do not use red and watch for much contrast between the colour you print and the colour of the paper.

3.3 Coding the numbers

You will perhaps not want to write something as ugly and error-prone as `{\eanfont2␣+AJjjJd-KTSMPN+}` manually. So you have to use a preprocessor[1]. With `wlean.mf` there comes a tiny Perl program (`codean.pl`) that does preprocessing within your TEX sources[2].

The TEX file to be filtered may contain any number of lines that have one of the following commands starting at the leftmost position.

- `\ean{`*12 or 13 digit number*`}` The number will be coded as EAN. If it is only 12 digits long, the checksum will be calculated, too.
- `\embed{`*12 or 13 digit number*`}` The number is used as a base for embedding article numbers &c. within an EAN.

[1] See bcfaq.tex for TEX code to go without preprocessing. It is very fine.
[2] codean.pl in the meantime also handles code 128. See bcfaq.tex

- `\eean{`*number with at most 11 digits*`}` This number is to be embedded within an EAN.
- `\isbn{`*valid ISBN*`}` An ISBN to make an embedded EAN of.

Let's look at an example: You want the ISBN `0-201-13448-9` to be embedded. So you write `\isbn{0201134489}`, but you might also use the embedding method and write `\embed{9780000000000}` and, somewhere later in the file, `\eean{020113448}`. In this latter case you have to omit the last digit, as ISBN loose their check digit in favour of the EAN check digit. Anyway you do it, you get your command replaced by `\EAN{`*13-digit-number-coded-strange*`}` in the output file.

But embedding is especially useful if you also write the program that reads the barcodes. This program might then extract your article number from an EAN starting with `20`, eg.

What you have to do is, of course, to use stub definitions for the three macros mentioned above—as they shall never actually be typeset—and to use a valid definition for `\EAN`. Then you run your TEX source through `codean.pl`. This program takes as first parameter the name of your original file and as second parameter the name of your destination file. If you omit the parameters, you will be asked for them.

You may of course also peek the source of `codean.pl` to see how EAN checksums are calculated, and so on.

3.4 Example

Here is a full example. Use this TEX source:

```
\font\eanfont=WLEAN
\def\ean#1{\message{Call codean.pl}}
\def\eean#1{\message{Call codean.pl}}
\def\isbn#1{\message{Call codean.pl}}
\def\embed#1{}
\def\EAN#1{\vbox{\vskip10pt\eanfont#1\vskip10pt}}
 Now, something to do:
\ean{4025700001030} % or, without checksum:
\ean{402570000103}
\embed{2500000000000}
\eean{123}
\isbn{0201134489}
```

Having run your file through `codean.pl`, the lines after the percent sign look like this:

```
 Now, something to do:
\EAN{4 +AcFHaa-KKLKNK+} % or, without checksum: %(4025700001030)
\EAN{4 +AcFHaa-KKLKNK+} %(402570000103)
```

```
\embed{2500000000000}
\EAN{2 +FAaaAa-KKLMNT+} % embedded(123)
\EAN{9 +HiaCaB-LNOOSN+} % ISBN(0201134489)
```

Running this file through TEX, you get DVI output containig EAN barcodes. Perhaps you wonder why there is not even a single EAN contained within this documentation. The reason is quite simple: You should be able to read the docs *before* you have installed the font. But now is the right time to try the example on your own. Better yet, you may code an EAN where you have taken the number from something like your favourite candy and then, having printed it, you may compare the bars. This is a nice way to spend your evenings. I actually started deciphering EAN codes in this way, several years ago.

4 The End

4.1 Address

Just in case you want to write to me, here is my address—but please note: I am not the EAN guru.


```
Peter Willadt
75177 Pforzheim
Germany
email: Willadt@t-online.de
```


I also would appreciate if only one version of the material contained in this package is distributed. So if you have any corrections, suggestions, &c., please do not hesitate to send them to me to incorporate them within this package.

4.2 Acknowledgement

I want to express my special thanks to Barbara Beeton for proofreading and making valuable suggestions. If there are still any typos or illegibilities, that is due to the fact that I had to change some things later on.

References

[1] Peter Olšak. The EAN barcodes by TEX. *TUGboat*, 15(4):459–464, 1994.