How to Combine Multiple Languages, PostScript and LaTeX

Timo Knuutila
Dept. of Computer Science
University of Turku
Lemminkäisenkatu 14 A
SF-20520 Turku
Finland

Abstract

A solution on how to handle multiple languages — even the accented ones — with correct hyphenation in standard LaTeX (with the Mittelbach–Schöpf font selection scheme) is presented. Moreover, the solution proposed makes it possible to easily switch between different languages and font families within the same document.

Introduction

Many of us are eagerly waiting for a new LaTeX with support for multi-lingual styles and international characters. There have already been some signs of the shape of things to come: the latest updates for LaTeX 2.09 have made it easier to integrate standard LaTeX with the `babel` style option of Johannes Braams (1991), and the new font selection scheme (`NFSS`) (Mittelbach and Schöpf, 1989) is accompanied with styles (currently in beta-test) to use the pre-release of the EC fonts defined in the Cork meeting (Ferguson, 1990) (the `DC` font family). However, while waiting for the official releases, the casual TeXnician has to manage somehow — usually with his own solutions.

This article concentrates mainly on the *interface* between LaTeX, international characters and languages using these. The language-specific adjustments used at higher levels (e.g., the name of "chapter") can be done with the `babel` styles. We begin the story with a section describing the reasons I undertook this effort. The current solution to the problems encountered is then presented in more detail.

Working with an Accented Language

In theory. When typing a TeX file, the typist thinks of a *glyph* (shape of a character or a symbol) and hits an appropriate key or key combination. The glyph becomes a small integer number, say i, in the file produced. For example, when working on a PC-compatible computer, the glyph-integer mapping is often defined by the IBM code page 850 (IBM coding hereafter). TeX reads the file and possibly converts i to another code j, which is then used with the precompiled hyphenation patterns. The resulting `dvi` file then contains integers j serving as pointers to glyphs in a font table. This glyph is finally made visible by the dvi driver that produces the raster image associated with the output code j.

In practice. A Scandinavian typist wants to have the glyph **ä** in his/her TeX text. He/she is forced to continuously type the sequence `\"a` instead of a single key, since the integer number produced by the key **ä** is not found from TeX's font tables, and neither is the character itself. The glyph is then constructed from two subparts: the accent ¨ and the letter **a**. One could perhaps stand this inconvinience, but accents cause an intolerable difficulty: TeX will not automatically hyphenate words containing accents — we have to write explicit discretionary breaks for them.

Modified versions of TeX do exist, like INRSTeX and MLTeX, which have the ability of hyphenating accented words, but they just are not TeX. In order to make TeX hyphenate properly words containing accented letters we have to create font tables for their representative numbers. However, the mapping from codes above 127 to glyphs (and vice versa) is *not* uniquely defined: there exist (to mention a few) 8 different ISO standards (Latin-1,...,Latin-8), numerous code pages from computer manufacturers, and the extended TeX font encoding scheme (Ferguson, 1990). The last scheme will most probably be accepted as a future standard, but many users of TeX feel reluctant to adopt it because of its incompatibility with the current TeX encoding.[1]

As the EC fonts have not yet been officially released, we describe a slight modification, which uses the Latin-1 coding for the 'upper half' of a 256-letter alphabet, but is downward compatible with current TeX documents. This encoding is

[1] The Greek letters do not reside in text fonts.

referred as the *Extended Computer Modern* (XCM).[2] These XCM fonts are totally based on the *virtual fonts* (Knuth, 1990) and the standard CM family — no extra METAFONT sources are used. The use of virtual fonts is further justified by the sheer size of a set of new raster fonts for a 256-letter alphabet and all magnifications (or true sizes).

PostScript. The distribution of the dvips driver is accompanied with a program, afm2tfm, which is able to translate the Adobe font metric files (afm) to TEX font metric (tfm) and virtual font (vf) files. Given an afm file as input, afm2tfm creates a font metric file for the raw PostScript characters (no character remapping) and a virtual font property list (vpl) file which defines the standard 7-bit character set using TEX's internal coding. The characters, whose codes are above 127, are mapped to their corresponding Adobe positions. Needless to say, this coding is different from both the Latin-1 coding and from the IBM coding.

We thus have three different codings for a single letter: one for input and two for the output. Because of two output codings, the hyphenation patterns create a further complication due to the need for different \language for Finnish in Computer Modern and for Finnish in PostScript. The only difference in the patterns of these 'languages' are the places where the (codes of) characters **ä** and **ö** are used.

Style files for PostScript. The new font selection scheme makes the definition of the PostScript fonts particularly simple since they are all generated by scaling the same font metric file. There currently exists many LATEX styles illustrating this ease.[3] We have used as a starting point psfonts.sty written by Dick van Soest.

Documents should be written by using only a few different font families — just recall the guidelines for these Proceedings. As an example, we could use Times Roman for plain text, Helvetica for sans serif and Courier for typewriter. Thus, it is not practical to always define all the different PostScript font families, regardless of what fonts are actually used. Therefore, the fonts should be declared *on demand*. The current implementation of NFSS makes the on-demand definition hard or impossible, because all the font declarations have to be made in the preamble of the document. Thus, one should either hack NFSS or explicitly define the fonts to be used in a style file. We have chosen the latter approach, but in order to keep the number of different style files minimal, one file contains all the commands needed to define the different families. The actual definition is made by a single command in the preamble.

Another reason to have one's own style for PostScript is the intermixing of PostScript, multiple languages and standard TEX fonts. Suppose that one writes documents in English and Finnish, both with Computer Modern and New Century Schoolbook (I do it all the time). Since it is faster to process dvi files containing real rather than the virtual fonts, we should use the CM fonts for English, and switch to XCM only when necessary, i.e., when changing the language. This approach also has the extra advantage of compatibility: the dvi files created for documents not using the language switching capabilities remain the same. However, when the XCM encoded PostScript fonts are used, no switching has to be done. Thus, the language switching device must be aware of the font family currently used.

The Evolution of MLATEX

How XCM was chosen. My first attempt to make TEX hyphenate Finnish text (this happened when TEX was younger than three years) was to create the extra glyphs by modifying the METAFONT sources of the Computer Modern font family, replacing some of the Greek characters with the new glyphs, and adjusting the hyphenation patterns according to the coding. Although the solution worked, it was clear that it was a non-portable hack; moreover, the hand-editing and testing of the METAFONT files was very time-consuming — especially for a person who does not know a bit about the programming language used!

Since I was working on a PC-compatible computer, the glyph-integer mapping was defined by the IBM code page 850. These integers (e.g., ^^84 for ä) were then mapped via TEX macros to the codes in the font tables. Knuth actually suggests ligatures (defined in the font property list file) to do the job (*The TEXbook* p. 46). However, this attempt implies key sequences such as a" should be used, which is not what was wanted.

Then came TEX 3.0 with 8-bit input and the facility of virtual fonts (Knuth, 1989). An instant idea was to exploit these facilities and create a virtual font on the top of each TEX font by adding the Scandinavian letters somewhere above the 7-bit barrier. At that time it seemed natural to use the

[2] It is an unfortunate coincidence that the Russian TEX project has adopted the same naming for their Cyrillic fonts (Malyshev et al, 1991).

[3] The definition of the CM fonts LATEX uses would be almost as easy if one used the sauter fonts (available from ftp.cs.umb.edu).

IBM coding for the foreign letters, because no extra transformations were then needed to process the documents. The process itself was straightforward: take a `tfm` file, convert it to a property list file (`pl`) with `tftopl`, hand-edit the result and create the `vpl` file, and finally run `vptovf` which gives the `vf` and `tfm` files needed. The hand-editing could have been done automatically, but I did not have the time (and skill?) to construct a program to do it.

As time passed, I became aware that there is a standard for the 8-bit character codes, namely ISO8859-1 also known as Latin-1. Tor Lillqvist (`tml@tik.vtt.fi`) from the Finnish Technical Research Center had created a set of virtual fonts, Extended Computer Modern, which used this standard. Actually, these `XCM` fonts do not contain all characters of Latin-1. Only the ones that can be constructed from the accents and characters of Computer Modern are included.[4] Moreover, the letters æ, œ and ß are located in their traditional places.[5] Tor had also written an Emacs Lisp macro which extends any property list file of `CM` fonts to a virtual property list file using the aforementioned `XCM` coding. The `XCM` fonts, which contain the standard font family of TeX, are created with this program.

Mapping input to `XCM`. The following problem was to remap the IBM codes to the `XCM` codes, but this was easily solved by declaring the extra characters as active. For example, the following definitions were needed for the letter ä:

```
\catcode`^^84=\active
\let^^84=^^e4
```

Later, I learned to utilize the TeX code page utility (`TCP`) of the emTeX-package (Mattes, 1990), and these kinds of macros were no longer needed.[6]

In 7-bit editor environments (e.g., a UNIX workstation and Gnu Emacs) there were two possibilities. In the first one the text is edited on a personal computer and then sent to the mainframe to perform the TeX-processing (and possibly other duties such as spell-checking). The character codes are mapped as explained above with a style file. Another possibility is to write documents as usual (using accents) on the workstation and redefine TeX's accents according to the `XCM` coding, again in a separate style file. As an example, `\"a` is translated to `^^e4` by the following redefinition of the ¨-accent:

```
\gdef\"##1{%
   \if##1a{^^e4}%
   %similar \if's for other
   %"-accented characters
   \else {\accent"7F ##1}\fi}
```

Actually, there is also a third way to write Finnish documents. The Scandinavians are used to replacing their national characters with letters found on a US keyboard. For example, `{` stands for ä, `|` stands for ö, etc. There exists a Scandinavian LaTeX, SLaTeX, designed with this coding in mind. Our system supports this implementation by mapping the SLaTeX characters to the `XCM` coding via a macro file.

Unifying PostScript and `XCM`. The first attempt to unify the PostScript and `XCM` codings was based on macros and character remapping: whenever PostScript fonts were used, each `XCM` code of a Scandinavian letter was defined to be active and mapped to the Adobe coding. Each PostScript font family (e.g., Times Roman) was associated with a LaTeX environment in which this mapping took effect. In 7-bit environments the accents had to be mapped similarly to their correct values.

Analogous to the character mappings, whenever a transition from the usual TeX coding to the Adobe world was made, the `\language` had to be adjusted, too. The resulting system, though working, was apparently quite clumsy and space-wasting.[7]

In order to have only one (`XCM`) set of hyphenation patterns for each language used, the Adobe codes had to be mapped somehow to their Latin-1 codes. The best solution was to modify `afm2tfm` in such a way that it mapped the characters (above 127) in the `vpl` file to their `XCM` codes, not to the Adobe ones. Actually, only a very slight modification to this program was needed. It should be noted that the latest version of `afm2tfm` (v7.0) (distributed with `dvips` v5.485 and up) allows different encodings for PostScript fonts. However, this facility seems to be still under developement and "for wizards only".

An Overview of the MiLaTeX System

The MiLaTeX system consists of the following parts:

[4] The `DC` fonts contain a full implementation of Latin-1.

[5] The PostScript fonts created by `afm2tfm` locate these letters similarly by default.

[6] The code page facility of emTeX can be used to map character codes to others before they are passed to TeX's mouth.

[7] Fortunately, the Finnish language is very orthogonal with regard to hyphenation: the pattern file takes less than 4 kilobytes in ASCII. The situation is much worse for more complex languages such as German, which needs over 40 kbytes of hyphenation patterns.

- files needed to create a new LaTeX format,
- LaTeX styles for multiple languages and PostScript fonts,
- XCM encoded virtual fonts for Extended Computer Modern and PostScript, and
- miscellaneous utility programs and files.

The main design principle has been *compatibility* with existing styles (babel, SLaTeX) and documents; no changes are needed for old LaTeX documents written in English. In addition, the dvi files for old documents remain identical.[8]

MLaTeX format files. The files used in the construction of the MLaTeX format can be divided in two parts: the font definition files (for NFSS) and the files needed for hyphenation. We currently have two replacements for fontdef.tex: fontdef.xcm and fontdef.xcm-sauter.[9] They both define a corresponding XCM font family and shape for each CM font (excluding the symbol fonts, of course).[10] The difference between these files is that fontdef.xcm-sauter uses true point sizes whereas fontdef.xcm uses the traditional scalings. For example, cmcsc20 is used instead of cmcsc10 at 20.74pt.

The hyphenation files contain a replacement for the master hyphenation file of LaTeX, lhyphen.tex, named lhyphen.mlatex. It defines the languages used and loads their hyphenation patterns. The languages are named in the form l@language, e.g., l@english, for compatibility with the babel styles. Currently, there are patterns for English, Finnish, German and Swedish. The pattern files are named hyphen.english, etc.

Files latin-1.tex and latin-1-xcm.tex contain the definitions for the \charcode, \uccode and \lccode of each Latin-1 character. In the latter file, only the characters contained in the XCM fonts are considered. It should be noted that latin-1.tex is of use with *any* implementation using Latin-1 encoded fonts (such as the DC fonts). The master hyphenation file loads either one of these files before the hyphenation patterns. Currently, latin-1-xcm.tex is the default. The essential contents of latin-1.tex are shown below.

```
\begingroup% save counters
% Make a loop from #1 to #2,
% change case at #3.
\def\setrange#1#2#3{%
\newcount\isochar%
\newcount\casechar%
\isochar=#1%
\loop%
  %We are in a group, hence global.
  \global\catcode\isochar=11%
  \casechar=\isochar%
  \ifnum\isochar<"#3% uppercase?
     \advance\casechar by "20%
     \global\lccode\isochar=\casechar%
     \global\uccode\isochar=\isochar%
  \else%
     \advance\casechar by -"20%
     \global\lccode\isochar=\isochar%
     \global\uccode\isochar=\casechar%
  \fi%
  \advance\isochar by 1%
\ifnum\isochar<#2\repeat}%
%
\setrange{"80}{"BD}{"A0}%
\setrange{"C0}{"FF}{"E0}%
\endgroup%
```

Style files. The styles are named multi (for *multiple* languages) and ps-nfss (for PostScript fonts with NFSS). Loading the multi style defines a macro \set<language> for each language defined in lhyphen.mlatex[11] (for example, \setswedish). The default language is English.

Style ps-nfss provides two commands for each PostScript font to be used — one for the declaration and one for the actual use. Fig. 1 contains a table of the commands currently defined.[12] Each font used in the document must be declared with a \load command in the preamble. However, Courier and Helvetica fonts are always defined, since they are commonly used for \tt and \sf. For changing back to Computer Modern, the command \computermodern is provided.

Both the commands for changing the font family and the language are designed to work properly with grouping i.e., they are in effect only within the group they are actually used. Below is an example on the use of these styles.

[8] It would be technically easier to always use the XCM fonts regardless of the language used, but then we would lose this property.

[9] Different names must be used for systems with a restricted length for file names (e.g., MS-DOS).

[10] The AMS symbol fonts, Euler fraktur fonts and the Cyrillic fonts from the University of Washington are declared, too. The user may comment out these declarations in the event these fonts are not needed.

[11] If the user of this system modifies the master hyphenation file by adding and/or removing languages, the style file should be edited, too.

[12] The lengthy names have been chosen on purpose in order to inhibit wild font abuse.

Font family	Loading	Using
Avantgarde	\loadavantgarde	\avantgarde
Bookman	\loadbookman	\bookman
Courier		\courier
Helvetica		\helvetica
NCS	\loadnewcentury	\newcentury
Palatino	\loadpalatino	\palatino
TimesRoman	\loadtimesroman	\timesroman
ZapfChancery	\loadchancery	\chancery

Figure 1: PostScript fonts supported by the ps-nfss style.

```
\documentstyle[multi,ps-nfss]{article}
\loadtimesroman\loadavantgarde
\timesroman
\begin{document}
This text is output in Times Roman.
{\avantgarde
This text is output in Avantgarde.
\setfinnish
Here we are, trying to hyphenate
English with Finnish patterns.
But still Avantgarde.
\computermodern
We won't give up trying to hyphenate
English with Finnish patterns.
Extended CM is used.}
English text, output in Times Roman.
\end{document}
```

Font files. The fonts consist of virtual font files and font metric files both for the extension of the standard LaTeX set of CM fonts and for the raw PostScript fonts distributed with the dvips driver. The XCM fonts contain the font families corresponding to the CM families cmr, cmti, cmsl, cmcsc, cmbx, cmbxsl, cmbxti, cmss, cmssbx, cmssi and cmtt with point sizes 5 – 12,14,17,20 and 25. One can always create his/her own new virtual font from an existing tfm or afm file by using the utility programs extend-cm.el and afm2tfm-iso described in the sequel. The tfm and vf files for the non-raw PostScript fonts found in the dvips package should be replaced with the new ones.

Utility files. The following is a brief description of the miscellaneous utility files. Some of them are usable only with a certain style package or with a specific TeX implementation.

mapacc.sty: A style file which redefines TeX's accents and maps them to the XCM character codes.

afm2tfm-xcm.c: A modification of afm2tfm using the XCM instead of the Adobe coding.

extend-cm.el: An Emacs Lisp macro package which can be used to create an extended vpl file from an existing pl file.

850-xcm.txt: emTeX code page for the translation from the IBM code page 850 to the XCM codes.

ibm2iso.tex: A macro file defining a mapping from the IBM code page 850 to the XCM codes.

ml-swedish.tex: A macro file to be used with the SLaTeX package (a replacement for swedish.tex).

ml-babel.hyphen: A multi-aware master hyphenation file for use with the babel styles. A similar ml-babel.switch is also provided.

Technicalities of the Style Files

We explain in more detail the implementation of the most important commands of multi and ps-nfss. The central thing is the co-operation of these styles. Before that, let us describe the stand-alone commands of ps-nfss.

Suppose that the command

```
\declare@psfont#1#2#3#4
```

defines the font family #1 with shape #3 and series #4 by using the font metric file #2.tfm for all standard LaTeX magnifications. Then the macro \declare@std below can be used to declare all the standard shapes and series for a usual PostScript font.

```
% declare@std{family}{normal}{italic}
% {slanted}{smallcaps}{bold}
\def\declare@std#1#2#3#4#5#6{%
  \declare@psfont{#1}{m}{n}{#2}%
  \declare@psfont{#1}{m}{it}{#3}%
  \declare@psfont{#1}{m}{sl}{#4}%
  \declare@psfont{#1}{m}{sc}{#5}%
  \declare@psfont{#1}{b}{n}{#6}%
  \extra@def{#1}{}{}}
```

For example, the command \loadavantgarde is defined as follows:

```
\def\loadavantgarde{%
  %Prevent double declaration.
  \@ifundefined{ava@loaded}{%
    \declare@std{ava}{pagk}{pagko}%
                {pagdo}{pagkc}{pagd}%
    \def\ava@loaded{}}{}}
```

When a PostScript font is to be used, we usually set the \sfdefault to Helvetica, \ttdefault to Courier and bfdefault to b. [13] This is done by macro \set@defs. The command go@PS#1 shown below switches to a given PostScript font family:

[13] For the other default values, we use the ones defined by NFSS.

```
\def\go@PS#1{%
  \let\in@ps=1%
  \def\default@family{#1}%
  \def\rmdefault{#1}%
  \set@defs%
  \fontfamily{#1}\selectfont}
```

The previous macro is then used to implement most of the font change commands, for example:

```
\def\avantgarde{\go@PS{ava}}
```

The two styles are aware of each other by checking the existence of certain macros. The style multi defines a macro `in@xcm`, which indicates whether we are currently using a language exploiting the `XCM` fonts or not. Similarly, ps-nfss defines an indicator `in@ps` which tells if we are currently using a PostScript font family.

These flags are used in the following two situations.

1. Changing a language may cause a switch between `CM` and `XCM`, but this kind of action is not needed when a `XCM` encoded PostScript font is used. Thus, the `\set` commands check whether the ps-nfss style is loaded and, if so, the value of `in@ps` tells the current situation.

2. The command `\computermodern` switches to `CM` or `XCM` depending on the current language. However, if the multi style is not loaded, we always change back to `CM`.

The implementations of the commands `\computermodern` and `\setfinnish` are given below. Commands `go@cmr`, `set@cm` and `set@xcm` set the default values likewise to `\go@PS`.

```
\def\setfinnish{%
   \set@language{\l@finnish}{2}{2}%
   \let\in@xcm=1%
   \@ifundefined{in@ps}{\set@xcm}{%
      \if0\in@ps\set@xcm\fi}}%
\def\setenglish{%
   \set@language{\l@english}{2}{3}%
   \let\in@xcm=0%
   \@ifundefined{in@ps}{\set@cm}{%
      \if0\in@ps\set@cm\fi}}%

\def\computermodern{%
   \let\in@ps=0%
   \@ifundefined{in@xcm}%
      {\go@cmr}%
      {\if0\in@xcm\set@cm%
        \else\set@xcm\fi}}%
```

The actual change of the language is accompanied by the setting of the left and right `hyphenmin`s (see below). It is the author's opinion that the values 2 and 3 are not the best possible for all languages on earth.

```
\def\set@language#1#2#3{%
  \language#1%
  \lefthyphenmin=#2%
  \righthyphenmin=#3}
```

Conclusion

We have presented an extension of LaTeX capable of handling multiple languages and PostScript fonts. The system is compatible in many ways, and even an English-speaking (and writing) LaTeX user might be tempted to install and use it. This is because the files for the `XCM` fonts do not take much space,[14] and the old documents do still turn into the same `dvi` files they used to. Yet, changing to international and/or PostScript is just a flip of a switch.

The development of a fast multi-lingual LaTeX (MLaTeX or not) would be much easier if we had the capability to load hyphenation patterns without `initex`. An even better solution would be to *precompile* the patterns (something like `\dumppatterns`) and then load and unload the precompiled hyphenation files dynamically. I also wish that we could adopt the `sauter` fonts as a standard for LaTeX. This would make the documents look better, and the definition and usage of all the `CM` fonts could be done in an orthogonal manner.

Bibliography

Braams, Johannes. "Babel, A Multilingual Style Option for Use with LaTeX's Standard Document Styles." *TUGboat* 12(2), pages 291 – 301, 1991.

Ferguson, Michael. "Report on Multilingual Activities." *TUGboat* 11(4), pages 514 – 516, 1990.

Knuth, Donald E. *The TeXbook*. 15th ed. Reading, Mass.: Addison-Wesley, 1989.

Knuth, Donald E. "The New Versions of TeX and METAFONT." *TUGboat* 10(3), pages 325 – 328, 1989.

Knuth, Donald E. "Virtual Fonts: More Fun for Grand Wizards." *TUGboat* 11(1), pages 13 – 23, 1990.

Malyshev, Basil, Alexander Samarin, and Dimitri Vulis. "Russian TeX." *TUGboat* 12(2), pages 212 – 214, 1991.

Mattes, Eberhard. *emTeX 3.0 User Manual*. 1990.

Mittelbach, Frank, and Rainer Schöpf. "A New Font Selection Scheme for TeX Macro Packages — The Basic Macros." TeX Users Group 10(2), pages 222 – 238, 1989.

[14] The total amount of space taken is about 330 kbytes.