

VFComb – a program for design of virtual fonts

A.S. Berdnikov and S.B. Turtia

Institute for Analytical Instrumentation
Rizskii pr. 26, 198103 St.Petersburg, Russia
berd@ianin.spb.su, turtia@ianin.spb.su

Abstract

The MS DOS program which enables to simplify the design of the virtual fonts is described. Its main purpose was to facilitate the integration of CM-fonts with cyrillic LL-fonts created by O. Lapko and S. Strelkov but it can be used for various applications. It uses the information from TFM-files (converted to ASCII form by TFtoPL) and the ASCII data files created by the User on its input, and produces the VPL-file on its output which can be converted to the virtual font using VPtoVF. The characteristic feature of the program is that it can assemble the font ligature tables and user defined ligature tables for the characters extracted from various fonts and combine the metric information from various TFM-files. VFComb supports the full syntax of PL-files and VPL-files as it was defined by D.E. Knuth and adds new commands like symbolic variables or conditional operators, which simplifies the creation and the debugging of the virtual fonts.

1 Introduction

This work has been inspired by the CyrTUG project to create the new cyrillic version of \TeX format files with the cyrillic fonts which are more close to the standards used in FSU than that created at Washington University (WN-family of cyrillic fonts). According to recommendations by D.E. Knuth [1] and following the experience of the other national TUGs the mechanism of the virtual fonts¹ has been selected as the base tool to add new (cyrillic) characters to already existing CM-fonts.

Unfortunately the software which was at CyrTUG's disposal at this moment was not well suited for this purpose. For the first version of new fonts the VF-files have been produced by the utility TFMerge 2.4 (IHEP \TeX ware, Protvino) with subsequent manual corrections, and the TFM-files have been produced by the separate METAFONT session

1. It is assumed below that you are familiar with the mechanism of the virtual fonts and with the paper [1]. If it is not so, it is highly recommended to read it before you proceed further.

which actually creates the real font containing both CM and LL characters [2]. The full process was time consuming and not effective.

The program VFComb described here has been designed to facilitate the process of combining real fonts into the virtual font and to make the iterations on creation and debugging of the final font more simple and less time consuming. The main points are:

- to get the metric information and the original ligature tables from the text VPL-files converted from the binary TFM-files by TFFtoPL utility;
- to add the user defined information which describes the relation between the characters from the virtual font and the characters from the real fonts;
- to add the user defined ligature tables which describe the ligature and kerning data for the pairs of characters corresponding to different real fonts;
- to create the VPL-file which combines all these data;
- to convert the ASCII VPL-file to the binary TFM-file and VF-file which forms the virtual font data.

The most of steps are performed automatically, and data files prepared by the User have flexible syntax so that just the same input files can be used to create full generic family of fonts rather than one single font.

2 Input and output data

The following input data must be supplied:

- the name of the virtual font (actually the name of the output VPL-file);
- the header data which is specified at the beginning of the virtual font;
- the assignment (if necessary) the numerical values to symbolic variables used in specification of other user defined data;
- the table(s) for mapping the characters of the real fonts into the characters of the virtual font;
- the additional ligature table(s) (if present) which contain the ligature and kerning information for the pairs of characters corresponding to different real fonts;
- the metric information, ligature and kerning tables, etc., for the characters of the real fonts;
- the options which control the process of creation of the virtual font.

The output of the program is the VPL-file which contains all information necessary to create the virtual font using the utility VPtoVF and the LOG-file which summarizes all printed messages and all operations performed by VFComb.

The output result depends on the mode of operation of the program. The mode is defined by the special `OPTION` data specified in input files. The most important options are connected with the algorithms of processing of ligature and kerning tables. the following options are available:

- to discard the ligature tables of the real fonts;
- to include in the virtual font the ligature tables of the real fonts;

- to include in the virtual font only those characters which are declared explicitly in user defined data, and discard the elements of the ligature tables which correspond to non-included characters of the real font;
- to add to the virtual fonts new characters not described in user defined data if these characters are described in ligature tables for already included characters, or they are connected in one chain with already included characters using specifications `NEXTLARGER` and `VARCHAR`.

These modes can be declared both as the global modes and as the modes for the particular real font.

All the files are the ordinary ASCII files which enables to control and to correct the operations performed by VFComb manually. To get the text files which describes the real fonts it is necessary to convert the corresponding binary TFM-files to ASCII form using the utility `TFtoPL`. The other way is to prepare these text files by hand following [1]. In this case it is possible that actually there is no real font corresponding to that used in virtual font. Although this situation looks strange, it enables to use the recursive tricks described in [1] where one virtual character refers to another virtual character of the same virtual font or of the other virtual font.

3 Mapping table and other user defined data

The files which contain the user defined data are specified in the command line using the parameter `/t:filename` when VFComb starts. It is possible to specify several data files using several parameters `/t` – in this case the data files are read one after another. It is also possible that the data file contains the explicit command to read another data file (like `\input` in `TEX` or `METAFONT`). The data files can be placed in the current directory, or can be specified using the full path name, or can be searched implicitly in one of the directories specified for VFComb as the source for missing data.

The user defined files can contain the following information:

- (`COMMENT ...`) – comment part which is skipped.
- (`VARIABLE ...`) – defines the symbolic names with some numerical values which are used later in user data specifications.
- (`OPTION ...`) – specifies various modes of combining real fonts into the virtual font.
- (`CHARACTER ...`) – the description of the virtual font characters. These tables can refer on the character from some real font or describe the sequence of DVI-commands corresponding to virtual font character.
- (`LIGTABLE ...`) – defines the additional ligature and kerning data which is included into virtual font. These tables are prepared following the syntax of PL and VPL-files described in [1].
- (`DISCARD ...`) – describes the characters from the real fonts which are not included in the virtual font at any case.

- (MAPFONT ...) – defines the mode of operation with particular real font.

The following examples give the illustration of some possibilities:

- CHARACTER H OF (SELECTFONT D 0) (SETCHAR 0 37)) – the character hex(0F) of the virtual font is mapped to the character oct(037) of the real font with index 0 (the relation between the names and the indices of the real fonts is described later).
- CHARACTER 0 40 (SELECTFONT D 3) (SETCHAR C w) (DVI ...)) – the character oct(040) of the virtual font is mapped to the sequence of DVI-commands, but its size parameters are equivalent to that of the character “w” of the real font with the index 3.
- (CHARACTER D 128 (DISCARD)) – the character 128 in the virtual font is not used and moreover, VFComb cannot use this character code when it adds to the virtual font new characters not described explicitly in user defined data.
- (OPTION (INCLUDELIG)) – include in the virtual font the characters of the real fonts if these characters are encountered in the ligature table for some already included character or if they are connected in one chain with already included characters using specifications NEXTLARGER and VARCHAR.
- (DISCARD (SELECTFONT D 0) (SETCHAR D 7)) – do not include in the virtual font the character 7 from the real font 0 even when it is encountered in the ligature table of some already included character.
- (MAPFONT D 2 (NOINCLUDE)) – do not include in the virtual font the characters of the real font 2 if these characters are not described explicitly in user defined data.
- (MAPFONT D 1 (7BIT)) – include in the virtual font all the characters 0–127 of the real font with index 1 assigning them just the same codes in the virtual font if the different mapping is not described explicitly in user defined data.

Formally there is no distinction between the file with the mapping table and the file with the ligature and kerning data, and all data can be specified as one continuous stream. It is better to keep different data blocks in different files so that the same file can be used to create various virtual fonts like it is done with METAFONT source files.

4 Specification of the real fonts

The real fonts which are used to create the virtual font are specified in the mapping table and user defined ligature tables using symbolic numbers. The relation of the symbolic number and the real font name is performed in the command line when VFComb starts.

Each font is specified using the separate parameter */f:filename*. The first parameter */f* corresponds to *zero* real font, the second parameter */f* corresponds to the first real font, etc. The number of real fonts specified in the command line is to be not less than the number of font indices used in the mapping table.

Each specification of the font name */f* causes the program to search for the corresponding PL-file (the ASCII file converted from TFM-file using the utility TFtoPL) and to load the font metric and ligature information containing in it. The font name inserted

in virtual font specification is the name of the file minus the extension and the path information.

It is a rare chance that it is necessary to modify the original font information – in most cases it is used “as it is”. Nevertheless to get full compatibility with the syntax of VPL-files [1] the font can be followed by the parameter */a* which specifies its magnification (it corresponds to the parameter `FONTAT` in VPL-file) and by the parameter */d* which specifies the explicit search directory (it corresponds to the parameter `FONTAREA` in VPL-file). The details can be found in VFComb manual and in [1].

5 The header of the virtual font

The header of each \TeX font (i.e., the header of corresponding TFM-file) contains the following parameters [1]:

`HEADER`, `CHECKSUM`, `SEVENBITSAFEFLAG`, `DESIGNSIZE`,
`DESIGNUNITS`, `CODINGScheme`, `FAMILY`, `FACE`,
`FONTDIMEN` and its sub-parameters (`SLANT`, `SPACE`, `QUAD`, etc.).

By default all header parameters of the virtual font except the parameters `CHECKSUM`, `SEVENBITSAFEFLAG`, `HEADER` and `FONTDIMEN/PARAMETER` are copied from the header of the first real font. The User can change the index of the real font used for this purpose and/or specify new header parameters which are read from the external file using standard syntax of PL-files. The parameters `SEVENBITSAFEFLAG`, `HEADER` and `FONTDIMEN/PARAMETER` are always ignored with corresponding warning messages. The interpretation of the parameters `DESIGNSIZE` and `DESIGNUNITS` is more complex and is described in VFComb manual (fortunately in most cases these header parameters are not modified). All the other header parameters except `CHECKSUM` are copied directly to the header of the virtual font.

The parameter `CHECKSUM` is treated separately: it is ignored when read from the header of the real font, and it is inserted in the header of the output VPL-file when read from additional header file. Generally there is no reason to specify the value `CHECKSUM` explicitly because by default it is calculated automatically when converting VPL-file to VF-file and TFM-file. In most cases the `CHECKSUM` value specified by the User is wrong which causes the warning messages of DVI-drivers when the virtual font is used. The exception is the explicit specification of zero value by (`CHECKSUM D 0`) which switches off the check of the control sum value at all (the latter operation is potentially dangerous because the incorrect or damaged TFM-file of the virtual font cannot be eliminated).

6 Additional features

Some extensions of the syntax of PL and VPL-files are introduced to make the process of preparing the user defined data more simple, namely:

- symbolic names for numerical constants (operator `VARIABLE`);
- dynamic loading of new data files (operator `LOAD`);
- binary data "`B binconst`" and symbolic values "`V varname`" which substitute numerical constants like "`D dec-value`", "`H hex-value`", "`O oct-value`", etc., in PL-files and VPL-files and user defined data;
- pseudo-arithmetic expressions which substitute numerical constants in PL-files and VPL-files and user defined data;
- conditional operators `IF-THEN-ELSE` which enable to include or to skip some portion of the data stream depending on some conditions;
- operator `(END)` which stops the processing of the current data stream before *end-of-file* is encountered.

The following example shows how the value of the variable `CODE` defines which value is assigned to the variable `DIMCH` and what external data file is loaded:

```
(IF-EQ V CODE D 0)
    \quad (VARIABLE (REAL DIMCH R 0.31415))
    \quad (LOAD NEWTABLE.TBF)
(ELSE)
    \quad (VARIABLE (REAL DIMCH R 2.7182))
    \quad (LOAD OLDTABLE.TBF)
(ENDIF)
```

The first operator `(IF-EQ V CODE D 0)` performs the comparison of two numerical values: "`V CODE`" and "`D 0`". The first value is the value of the symbolic variable `CODE`, and the second value is the decimal integer 0. Depending on the result of comparison the real value 0.31415 or 2.7182 is assigned to the real variable `DIMCH`, and one of two files `newtable.tbf` or `oldtable.tbf` is loaded like their contents is typed at this place.

7 Conclusion

The program `VFCOMB` is written in Borland Pascal and is distributed "as it is" together with the source code and \LaTeX manual. The program as well as the manual is far from perfect state, and all suggestions about corrections and noticed errors are welcomed.

8 Acknowledgements

This research was partially supported by a grant from the Dutch Organization for Scientific Research (NWO grant No 07-30-007).

References

- [1] D. Knuth. Virtual fonts: More fun for grand wizards. *TUGBoat*, 11.1:13–23, 1993.
- [2] O. Lapko and S. Strelkov. `MAKEFONT` as a part of `CyrTUG-emTeX` package.