

**NAME**

**makejvf** – Make Japanese VF file from Japanese T<sub>E</sub>X TFM file

**SYNOPSIS**

**makejvf** [*<options>*] *<T<sub>E</sub>X TFM file>* *<PS font TFM>*

**DESCRIPTION**

**makejvf** is a tool to generate Japanese VF file from Japanese T<sub>E</sub>X TFM (JFM) file for use with dvips.

When processing Japanese texts, pT<sub>E</sub>X refers to JFM (min10.tfm, jis.tfm etc.), which includes definitions of some different character widths and metric glue/kerns. For most punctuations and quotation marks, the character widths are truncated to less than 1 zw (zenkaku-width; the width of ordinary Kanji characters), and metric glue/kerns are inserted as a substitute.

On the other hand, in Japanese PS fonts, all punctuations and quotation marks have the same character widths as ordinary Kanji characters. For this reason, when dvips processes the resulting DVI, these characters have to be shifted to the left by the amount of glue/kerns inserted.

To achieve this, Virtual fonts (VF) and PS TFM files are required; When VF contains the commands of shifting characters, PS font TFM can have the exact character widths of PS fonts.

The program **makejvf** can be used for this purpose. It inputs a pT<sub>E</sub>X JFM file (referred to as *<T<sub>E</sub>X TFM file>* in **SYNOPSIS** above), and outputs a corresponding VF file (with the same basename as *<T<sub>E</sub>X TFM file>*) and a JFM file for a PS font JFM file (*<PS font TFM>* above).

**OPTIONS**

- C** Condensed ("Cho-tai") mode.
- K** *<PS-TFM>*  
Map Kana (more exactly, non-Kanji) characters to another PS font JFM named *<PS-TFM>*.
- b** *<integer>*  
Base line shift amount; the *integer* represents a relative value, using the character height as a base of 1000. When a positive integer is specified, the characters are lowered. When a negative integer is specified, the characters are raised.
- m** Replace single/double quotation marks (', ') with single/double prime quotation marks (so-called "minute") in vertical writing. The replacement is realized by manipulating glyphs of prime and double prime (JIS 0x216C and 0x216D; Unicode U+2032 and U+2033), *not* by putting actual glyphs designed for quotation marks (Unicode U+301D and U+301E/U+301F).
- a** *<AFMfile>*  
Name of the input AFM file used for Kana-tsume mode. This option is unsupported.
- k** *<integer>*  
Kana-tsume (narrower spaces between Kana characters) margin amount; the *integer* represents a relative value, using the character width as a base of 1000. This option should be accompanied with **-a** option. This option is unsupported.
- i** Start mapped font ID from No. 0 in output VF (by default, makejvf defaults to No. 1).
- e** Enhanced mode; the horizontal shift amount is determined from the glue/kern table of input JFM file.

By default, **makejvf** uses the hard-coded value as the horizontal shift amount, which is (mostly) optimized for Japanese fonts. When enhanced mode (option **-e**) is enabled, the shift amount is determined from the input pT<sub>E</sub>X TFM (JFM) file, which is likely to output most suitable VF for the JFM.

For most standard Japanese JFM (like jis.tfm and its derivatives), the output VFs from both modes will have no significant difference. For simplified/traditional Chinese JFM (like upschrh.tfm and uptchrh.tfm), the output VF from enhanced mode will be better. For min10.tfm and its derivatives, enhanced mode should *never* be enabled, since the characterization in min10.tfm is non-standard.

**-t** <CNFfile>

Use <CNFfile> as a configuration file.

**-u** <Charset>

UCS mode. Available charsets are: gb (GB = Simplified Chinese), cns (CNS = Traditional Chinese), ks (KS = Korean), jis (JIS = Japanese), jisq (JIS quote only), custom (user-defined CHARSET from <CNFfile>; see **CONFIGURATION FILE FORMAT** section).

Options below are effective only in UCS mode:

**-J** <PS-TFM>

Map single/double quote to another JIS-encoded PSfont TFM.

**-U** <PS-TFM>

Map single/double quote to another UCS-encoded PSfont TFM

**-3**

Use set3, that is, enable non-BMP characters support (with UCS mode). By default **makejvf** does not output >=U+10000, to reduce file size and to avoid problems with old DVI drivers. Recent versions of dvipdfmx and others can handle VF with >=U+10000 (= set3 in DVI language), therefore **-3** might be helpful.

**-H**

Use half-width Katakana.

## EXAMPLE

If you want to use min10 as Ryumin-Light-H, run  
makejvf min10.tfm rml

This generates min10.vf and rml.tfm. Put these files in an appropriate directory under TEXMF tree, and add the following line to psfonts.map.

```
rml Ryumin-Light-H
```

## CONFIGURATION FILE FORMAT

With **-t** option, you can give **makejvf** a custom settings for generating VF. The syntax is:

```
% comment line
MOVE      <code> <right> <down>

REPLACE   <code> <new code>
CHARSET   <code>,<code>,<code>..<code>,<code>,
+         <code>,<code>..<code>
```

Each line should begin with a command, and should be TAB-separated. Line starting with % is a comment, and empty lines are ignored.

The **MOVE** command specifies horizontal/vertical shift amount for the individual character *<code>*. The **REPLACE** command replaces the character *<code>* with *<new code>*. The **CHARSET** command sets the custom character set of output VF; the + character continues from the previous line.

An example usage can be found in `uptex-fonts` project. See GitHub repository <https://github.com/texjporg/uptex-fonts>.

## SEE ALSO

More detailed description of **makejvf** in Japanese is available at `$TEXMFDIST/doc/fonts/ptex-fonts/README_makejvf`

## AUTHOR

This manual page was written by Japanese T<sub>E</sub>X Development Community <https://texjp.org>. For more information, see GitHub repository <https://github.com/texjporg/ptex-fonts>.

Many thanks to Atsuhito KOHDA [kohda@debian.org](mailto:kohda@debian.org), for providing another manpage in Debian GNU/Linux system.