The fixjfm package*†

Yue Zhang

2017-09-04 v0.4

Abstract

This package fixes several bugs in the JFM format. Both LaTeX and plain TeX are supported.

1 Introduction

The JFM format is an extension of the TFM format and is used for typesetting CJK characters with *pTeX. It works perfectly under almost all circumstances, but there are still at least two bugs:

- Bug 1: the indentation is incorrect if the first character of a paragraph is an opening fullwidth punctuation mark, see Figure 1;
- Bug 2: the spacing between two consecutive fullwidth punctuation marks cannot be adjusted if the font is changed there, see Table 1.

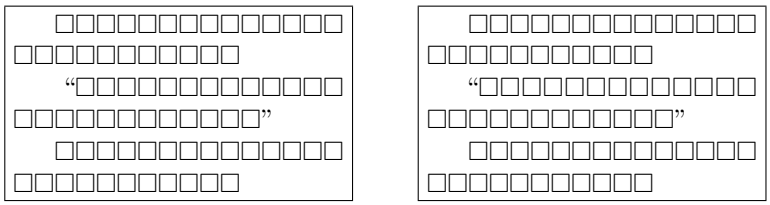

Figure 1: Comparison of the indentation produced without this package (left) and with this package (right)

Table 1: Comparison of the adjustment of the spacing between two consecutive fullwidth punctuation marks

Font change	This package	Input	Output
no	either no or yes	□□, “□□”	□□, “□□”
yes	no	□□, \textgt{ “□□” }	□□, “□□”
yes	yes	□□, \textgt{ “□□” }	□□, “□□”

It seems that macro is the easiest solution to these problems. However, the second bug can only be fixed when using ε-(u)pTeX (under most, but not all

*CTAN Homepage: https://ctan.org/pkg/fixjfm
†Repository: https://github.com/Man-Ting-Fang/fixjfm

circumstances), because the primitive `\lastnodechar` which is introduced by ε-pTEX is required. Anyhow, using this package with pTEX or any one of its derivatives does not hurt. Please also keep in mind that owing to technical limitations, this package is not a panacea.

There is also another improvement: `\leavevmode` is redefined as `\quitvmode` if the latter is available as a primitive (among all *pTEX engines to date, only ApTEX has `\quitvmode`). So after loading this package, you can always use `\leavevmode` and need not worry about `\quitvmode`. If you are wondering about what the difference between them is, please see *The pdfTEX user manual*.

2 Basic usage

This package has no options. It is recommended that this package should be loaded before any other packages. If you are using LATEX, load this package at the beginning of your preamble:

```
\documentclass...
\usepackage{fixjfm}
```

or even before `\documentclass` (use `\RequirePackage` instead):

```
\RequirePackage{fixjfm}
\documentclass...
```

If you are using plain TEX, put the following line near the beginning of your `.tex` file:

```
\input fixjfm.sty
```

After loading this package, the first bug mentioned above can be automatically fixed under most (but not all) circumstances. If you find that it cannot be fixed somewhere, you can add `\<` manually before the opening fullwidth punctuation mark which begins the paragraph.

In the case of the second bug mentioned above, the situation is different. If you are using LATEX, `\textmc` and `\textgt` are redefined by default so that the bug can be automatically fixed (again, under most, but not all circumstances). However, `\mcfamily` and `\gtfamily` remain unchanged. The difference here is similar to that between, say, `\textit` and `\itshape`: `\textit` automatically takes care of any necessary italic correction on either side of the argument, while `\itshape` does nothing about that. Just like `\itshape` and `\/`, you should add

```
\fixjfmspacing
```

by yourself after the font change that appears between two consecutive fullwidth punctuation marks. For example:

```
□□, {\gtfamily\fixjfmspacing “□□” }\fixjfmspacing “□□”
```

If you are using plain TEX, you should always add `\fixjfmspacing` by yourself, because plain (*p)TEX does not have `\textmc`, `\textgt`, or the like.

If you are using LATEX and prefer the standard version[1] of `\textmc` and `\textgt`, you can declare

[1] Since v0.3, the “standard version” is similar to jsclasses and BXjscls rather than (u)pLATEX.

```
\UseStandardCJKTextFontCommands
```

In contrast,

```
\UseFixJFMCJKTextFontCommands
```

redefines `\textmc` and `\textgt` and is declared by default. These two commands are important. Consider the following example:

```
\textgt{\Large □□}
```

It causes a fatal error when the fixjfm version of `\textgt` is used. The solution is to change `\textgt` back to the standard version, either globally or locally:

```
\UseFixJFMCJKTextFontCommands   \textgt{□□}
\UseStandardCJKTextFontCommands \textgt{\Large □□}
\UseFixJFMCJKTextFontCommands   \textgt{□□}
{\UseStandardCJKTextFontCommands \textgt{\Large □□}}
```

3 Advanced usage

The spacing between two consecutive fullwidth punctuation marks is produced by a glue item specified in the corresponding JFM file. However, if the font is changed there, the JFM format cannot work correctly, so this package puts another glue item there to adjust the spacing. The natural width of the new glue item is calculated according to the corresponding JFM file, so you need not worry about it. However, the stretch and shrink components cannot be extracted from JFM, so the following two commands are provided:

```
\SetFixJFMSpacingStretch
\SetFixJFMSpacingShrink
```

They expect an argument specifying how much space to stretch and shrink respectively. This package sets both of them to `0.05zw` by default.

Macro writers may want to (re)define commands like the fixjfm version of `\textmc` and `\textgt` for CJK text fonts, thus the following command is provided:

```
\DeclareFixJFMCJKTextFontCommand
```

It is similar to LaTeX's `\DeclareTextFontCommand`. For example, `\textmc` is redefined by this package as follows:

```
\DeclareFixJFMCJKTextFontCommand\textmc{\mcfamily}
```

and `\textgt` is similar. In contrast,

```
\DeclareStandardCJKTextFontCommand
```

(re)defines a CJK text font command like the standard version of `\textmc` and `\textgt`.

Finally, there are several macros that may be useful for TeXnicians:

```
\FixJFMSpacing
\fixjfmparindent
\FixJFMParindent
\EveryparPreHook
\EveryparPostHook
```

They are worthless for ordinary use. If you are interested in them, please have a look at the source code.

4 Compatibility

The LaTeX-only package bxjaprnind also focuses on the first bug mentioned above. It provides some additional functionalities and also supports some other TeX engines. This package can be used together with bxjaprnind, but please note that if you prefer bxjaprnind, you have to load it before this package, and vice versa, in case both of them are loaded.

5 Acknowledgements

The source code of this package is mostly taken from[2] (in alphabetical order):
ε-pTeX Wiki (Hironori Kitagawa):

`https://ja.osdn.net/projects/eptex/wiki/lastnodechar`

everyhook (Stephen Checkoway):

`https://ctan.org/pkg/everyhook`

jsclasses (Haruhiko Okumura et al.):

`https://ctan.org/pkg/jsclasses`

and platex (Kazuki Maeda & Japanese TeX Development Community):

`https://ctan.org/pkg/platex`

Many thanks to the authors of these packages.

6 History

2017-09-02 v0.2

- First public version.

2017-09-04 v0.3

- Fixes.
- Make `\UseFixJFMCJKTextFontCommands` and `\UseStandardCJKTextFontCommands` also available after `\begin{document}`.

[2] This package also improves the code slightly.

- Add `\DeclareStandardCJKTextFontCommand` and use it for redefining the standard version of `\textmc` and `\textgt`.

Thanks to Hironobu Yamashita for suggesting all these changes:

`https://github.com/Man-Ting-Fang/fixjfm/pull/1`

2017-09-04 v0.4

- Bug fix, thanks to Hironobu Yamashita:

`https://github.com/Man-Ting-Fang/fixjfm/pull/2`