

zhmCJK 宏包

刘海洋

leoliu.pku@gmail.com

2016/11/23 v0.9c

1 简介

zhmCJK 宏包是一个基于 zhmetrics 机制 [8] 和 CJK 宏包 [4] 的 CJK 文字配置宏包。宏包的名称 zhmCJK 就取自 zhmetrics 和 CJK 这两个包。它可以动态地设置 CJK 字体，仅指定字体文件名，就完成原来十分复杂的 CJK 字体安装设置工作。

zhmCJK 提供了尽可能简单的用户界面。除了提供对 CJK 宏包所用字体的实时安装设置功能（默认 UTF-8 编码），zhmCJK 还同时加载了 CJKpunct 和 CJKspace 宏包处理标点压缩和字符间距。

zhmCJK 支持 pdfTeX 和 DVIPDFMx 两种输出驱动，可以使用 pdflatex 或 latex + dvipdfmx 的方式编译。

2 用法

2.1 宏包载入与选项

只需要在导言区使用

```
\usepackage{zhmCJK}
```

即可载入宏包。

宏包可以带有一些选项（表 1）。例如如果要打开自动伪斜体，就可以用

```
\usepackage[AutoFakeSlant]{zhmCJK}
```

一般来说使用默认的设置，即不用任何选项即可。

选项	默认值	功能
<code>encoding=⟨编码⟩</code>	UTF8	设置 CJK 输入的编码，如 GBK, Bg5 等，参见 [4]。
<code>pdffakebold=⟨true false⟩</code>	true	设置伪粗体类型为 PDF 原语生成；若设置为 false 则沿用旧的 CJK 伪粗体机制
<code>AutoFakeBold=⟨数字⟩</code>	0.4	全局设置自动伪粗体的粗细，单位为 bp；若设置为 0 则禁用自动伪粗体。如果参数省略，取值为 0.4。
<code>AutoFakeSlant=⟨数字⟩</code>	0	全局设置自动伪斜体的倾斜度；若设置为 0 则禁用自动伪斜体。如果参数省略，取值为 0.167。
<code>cmap=⟨CMap 值⟩</code>	unicode	设置字体文件使用的 CMap 映射，对多数 CJK 字体可以使用默认值 unicode，相当于使用基于 UCS2 的字体编码；部分字体需要设置为 UniGB-UTF16-H 等映射。仅对 DVIPDFMx 有效。
<code>embed=⟨true false⟩</code>	true	设置字体是否嵌入。仅对 DVIPDFMx 有效。

表 1: zhmCJK 宏包选项

2.2 宏包提供的命令

zhmCJK 的基本用户界面与 xeCJK 宏包 [7] 十分相似，定义字体的几个命令语法大体相同。

```
\setCJKmainfont      \setCJKmainfont{⟨字体文件名⟩}[⟨选项⟩] 或
                     \setCJKmainfont[⟨选项⟩]{⟨字体文件名⟩}
```

设置正文默认罗马族的 CJK 字体，字体用 TrueType 文件名表示。该命令影响 `\rmfamily` 和 `\textrm` 的字体。

字体选项与宏包选项基本对应，可用的字体选项有：`BoldFont`, `ItalicFont`, `BoldItalicFont`, `SlantedFont`, `BoldSlantedFont`，分别用来设置不同字型的变体；`AutoFakeBold`，设置单个字体族的伪粗体；`AutoFakeSlant`，设置单个字体族的伪斜体；`encoding`，设置单个字体

使用的输入编码; `cmap`, 设置单个字体使用的 CMap 映射; `embed`, 设置单个字体是否嵌入。

`\setCJKromanfont` 是 `\setCJKmainfont` 的别名。
例如, 使用

```
\setCJKmainfont{simsum.ttc}[BoldFont=simhei.ttf,ItalicFont=simkai.ttf]
```

将使用文件名为 `simsum.ttc` 的字体 (中易宋体) 作为正文罗马族字体, 同时其粗体形式设置为 `simhei.ttf` (中易黑体), 其斜体形式设置为 `simkai.ttf` (中易楷体)。

又如, 使用

```
\setCJKmainfont{FandolSong-Regular.otf}
[cmap=UniGB-UTF16-H,
BoldFont=FandolSong-Bold.otf,
ItalicFont=FandolKai-Regular.otf]
```

可以设置 Fandol 字体作为正文字体。

`\setCJKsansfont` `\setCJKsansfont{<字体文件名>}[<选项>]` 或
`\setCJKsansfont[<选项>]{<字体文件名>}`

设置正文无衬线族的 CJK 字体。影响 `\sffamily` 和 `\textsf` 的字体。例如用

```
\setCJKsansfont{simhei.ttf}
```

将使用字体 `simhei.ttf` (中易黑体) 作为正文无衬线字体。

`\setCJKmonofont` `\setCJKmonofont{<字体文件名>}[<选项>]` 或
`\setCJKmonofont[<选项>]{<字体文件名>}`

设置正文等宽族的 CJK 字体。影响 `\ttfamily` 和 `\texttt` 的字体。例如用

```
\setCJKmonofont{simfang.ttf}
```

将使用字体 `simfang.ttf` (中易仿宋体) 作为正文等宽字体。

`\setCJKfamilyfont`

```
\setCJKfamilyfont{<族>}{<字体文件名>}[<选项>] 或
\setCJKfamilyfont{<族>}[<选项>]{<字体文件名>}
```

定义新的 CJK 字体族并指定字体。例如用

```
\setCJKfamilyfont{yahei}{msyh.ttc}[BoldFont=msyhbd.ttc]
```

可以定义名为 `yahei` 的 CJK 字体族，其对应的实际字体文件名为 `msyh.ttc`（微软雅黑），粗体形式是 `msyhbd.ttc`（粗体的微软雅黑）。

以上字体设置命令只能在文档导言区使用。中文字体可以用 `CJK` 宏包的 `\CJKfamily` 命令切换。例如用 `\CJKfamily{yahei}` 就可以选择前面定义的微软雅黑字体。

`\CJKrmdefault` 保存 `\rmfamily` 所使用的 CJK 字体族，默认值是 `rm`。类似西文字体的 `\rmdefault`。

`\CJKsfdefault` 保存 `\sffamily` 所使用的 CJK 字体族，默认值是 `sf`。类似西文字体的 `\sfdefault`。

`\CJKttdefault` 保存 `\ttfamily` 所使用的 CJK 字体族，默认值是 `tt`。类似西文字体的 `\ttdefault`。

`\CJKfamilydefault` 保存 `\normalfont` 所使用的 CJK 字体族，默认值是 `\CJKrmdefault`。类似西文字体的 `\familydefault`。例如，使用

```
\renewcommand\familydefault{\sfdefault}
\renewcommand\CJKfamilydefault{\CJKsfdefault}
```

可以将全文的 CJK 和西文默认字体改为无衬线字体族。

2.3 示例文件

```
1 (*test)
2 % UTF-8 编码
3 % 使用 pdflatex 或 latex + dvipdfmx 编译
4 \documentclass[12pt,a4paper]{article}
5
6 \usepackage{zhmCJK}
7 \setCJKmainfont{simsun.ttc}[BoldFont=simhei.ttf,ItalicFont=simkai.ttf]
8 \setCJKsansfont{simhei.ttf}[AutoFakeBold=0]
9 \setCJKmonofont{simfang.ttf}[AutoFakeSlant]
10 \setCJKfamilyfont{xinwei}{STXINWEI.TTF}
11
12 \title{\bfseries 一个华丽的 zhmCJK 测试}
13 \author{Leo Liu}
14 \date{2016 年 11 月 23 日}
15
16 \begin{document}
```

```

17
18 \maketitle
19
20 正文是宋体字, \textbf{加粗变成黑体}, \textit{意大利体变成楷体}。
21
22 \textsf{这是黑体字, \textbf{禁用自动加粗}, \textsl{默认不自动倾
    斜}}
23
24 \texttt{这是仿宋字, \textbf{默认自动加粗}, \textsl{打开自动倾斜}}
25
26 {\CJKfamily{xinwei}还有华文新魏}
27
28 \end{document}
29 </test>

```

3 安装

3.1 组成与依赖项

zhmCJK 源代码包包含如下文件:

- zhmCJK.dtx, zhmCJK.ins: 这是 zhmCJK 的 DocStrip 包与驱动。
- zhmCJK.lua: 这是用来生成 TFM 文件 [8] 和字体名映射文件的 Lua 脚本。
- zhmCJK-addmap.pl: 这是 TFM 字体映射的合并脚本。

除此以外, 使用 zhmCJK 还依赖:

- CJK, CJKutf8, CJKspace 宏包, 同属于 CJK 集合。
- CJKpunct 宏包。
- ifpdf, kvoptions 宏包, 同属于 oberdiek 集合。

编译使用 zhmCJK 的文档需要 ϵ -TeX 支持, 并要求使用较新版本的 pdfTeX 或 DVIPDFMx 生成 PDF 文件。

3.2 生成与安装

生成格式文件和文档:

1. 对 TeX Live 等发行版, 执行

```
texlua zhmCJK.lua map
```

生成:

- `zhmCJK.tfm`: 通用的 CJK 字符 TFM 文件, 可用于大部分常见的中文字体。
- `zhmCJK.map`, `texfonts.map.template`: 字体名映射文件。

对 MiKTeX 等发行版, 执行

```
texlua zhmCJK.lua nomap
```

生成目录 `miktex-tfm` 下的 32 个子目录 `zhm*`, 每个子目录下有 256 个 TFM 文件 `zhm**.tfm`。共 8192 个 TFM 文件 `miktex-tfm/zhm*/zhm**.tfm`。

2. 执行:

```
latex zhmCJK.ins
```

生成格式文件 `zhmCJK.sty` 和示例文件 `zhmCJK-test.tex`。

3. 执行:

```
latex zhmCJK.dtx
makeindex -s gind zhmCJK.idx
makeindex -s gglo -o zhmCJK.gls zhmCJK.glo
latex zhmCJK.dtx
latex zhmCJK.dtx
dvi2pdf zhmCJK.dvi
```

生成宏包文档。编译文档可以使用 `pdflatex` 命令。编译文档需要 T_EX 发行版安装有 Fandol 字体包。

设 `TEXMF` 是 T_EX 系统的 TDS 根目录, 按如下目录结构复制宏包的各个文件:

```
TEXMF/doc/latex/zhmCJK/zhmCJK-test.tex
TEXMF/doc/latex/zhmCJK/zhmCJK.pdf
TEXMF/doc/latex/zhmCJK/README
TEXMF/fonts/map/fontname/zhmCJK.map
TEXMF/fonts/tfm/zhmCJK/zhmCJK.tfm
```

```
TEXMF/source/latex/zhmCJK/zhmCJK.dtx
TEXMF/source/latex/zhmCJK/zhmCJK.ins
TEXMF/source/latex/zhmCJK/zhmCJK.lua
TEXMF/tex/latex/zhmCJK/zhmCJK.sty
```

并将 `texfonts.map.template` 内容合并到

```
TEXMF/fonts/map/fontname/texfonts.map
```

也可以运行 Perl 脚本 `zhmCJK-addmap.pl` 完成合并。复制完所有文件后运行 `texhash` 命令（可能需要管理员权限）刷新文件名数据库，宏包即可使用。

对于 MiKTeX，`texfonts.map` 和 `zhmCJK.map` 的映射机制无效，则需要复制整个 `miktex-tfm` 目录下的所有文件到：

```
TEXMF/fonts/tfm/zhmCJK/miktex-tfm/zhm*/zhm**.tfm
```

注：实际下载的安装包已经生成好上述文件和文档，可以不必手工生成。如果下载的是 TDS 安装包，则只需按目录结构解压复制，刷新文件名数据库即可。对于 MiKTeX 系统，需要把打包的 TFM 文件解压出来。

3.3 字体搜索路径

`zhmCJK` 可以使用 `kpathsea` 库 [2] 所能搜索到的 Unicode 编码 CJK 字体，包括操作系统安装的字体。`pdfTeX` 支持 TrueType 格式（扩展名 `.ttf`，`.ttc`），`DVIPDFMx` 还支持 OpenType 格式（扩展名 `.otf`）。

Windows 操作系统下，MiKTeX 和 `TeX Live` 可以不做特别的设置，即可正确搜索操作系统的字体目录。

在 Linux 操作系统下，需要设置 `TeX Live` 的 `texmf.cnf` 配置文件，在其中的 `OSFONTDIR` 变量中增加操作系统字体目录。例如 `TeX Live` 在目录 `/usr/local/share/texlive/` 安装，则新建（或修改）文件 `/usr/local/share/texlive/texmf.cnf`，在其中增加如下内容：

```
OSFONTDIR = /usr/share/fonts//;~/.fonts//
```

详情可参考 `TeX Live` 手册及原有 `texmf.cnf` 文件中的相关注释说明。

可以使用 `kpsewhich` 命令检查 `zhmCJK` 能否搜索到指定的字体文件。例如在 Windows 系统中使用命令：

```
kpsewhich simsun.ttc
```

应该得到输出：

`c:/Windows/fonts/simsun.ttc`

对非 Windows 系统，文件和路径区分大小写。

4 已知问题

- 受制于预定义的映射文件 `texfonts.map`，使用 `zhmCJK` 在同一文档中能够使用的字体文件是有限的。目前只有 32 个。
- 在 MiKTeX 中 `texfonts.map` 映射无效，无法使用单一 TFM 文件处理所有的 CJK 字体，因此需要生成所有对应的 TFM 文件才能使用。

5 代码实现

5.1 字体设置包 zhmCJK.sty

```
30 <*package>
```

首先进行编译引擎测试。

`\ifzhm@primitive` 测试是否为原始控制序列，用于判断编译引擎。代码参考 `ifetex` 宏包，有改动。

```
31 \def\ifzhm@primitive#1{\begingroup
32   \edef\tempa{\meaning#1}%
33   \edef\tempb{\string#1}%
34   \expandafter\endgroup
35   \ifx\tempa\tempb}
```

需要 ϵ -TeX 支持。对非 ϵ -TeX 引擎报错并退出宏包。

```
36 \ifzhm@primitive\eTeXversion\else
37   \PackageError{zhmCJK}%
38     {You are not running an eTeX-compatible engine.
39     zhmCJK needs an eTeX-compatible TeX engine}%
40     {If 'latex' and 'pdflatex' commands fail,
41     try 'elatex' or 'pdfelatex' command.
42     If all fail, you may need to upgrade your TeX system.}
43   \expandafter\endinput
44 \fi
```

不支持 LuaTeX。遇到 LuaTeX 引擎报错并退出宏包。

```
45 \ifzhm@primitive\luatexversion
46   \PackageError{zhmCJK}%
47     {You are wrongly using lualatex or dvilualatex to compile
48     the file. zhmCJK does not support LuaTeX engine.}%
49     {Use pdflatex or latex+dvipdfmx to compile the file.}
50   \expandafter\endinput
51 \fi
```

不支持 XeTeX。当发现使用 XeTeX 时报错，并载入 `xeCJK` 宏包再退出。

```
52 \ifzhm@primitive\XeTeXrevision
53   \PackageError{zhmCJK}%
54     {You are wrongly use xelatex to compile the file.
55     zhmCJK do not support XeTeX engine.}%
56     {You can use xeCJK instead.}
57   \RequirePackage{xeCJK}
```

```
58 \expandafter\endinput
59 \fi
```

载入工具宏包。

```
60 \RequirePackage{ifpdf}
61 \RequirePackage{kvoptions}
62 \SetupKeyvalOptions{
63   family=zhm@opt,
64   prefix=zhm@}
```

cmap 定义使用的 DVIPDFMx 字体映射所使用的 CMap 映射，默认是 `unicode`，即使用 UCS2 编码的字体映射。多数 CJK 字体可以使用默认值，但少量需要单独设置才能使 DVIPDFMx 工作正常。

```
65 \DeclareStringOption[unicode]{cmap}
```

embed 控制字体是否嵌入。只对 DVIPDFMx 有效。

```
66 \DeclareBoolOption[true]{embed}
67 \AddToKeyvalOption*{embed}{%
68   \ifpdf
69     \PackageWarning{zhmCJK}
70       {The ‘embed’ option is not supported by pdfTeX driver.\MessageBreak
71         Ignoring option ‘embed’}
72   \fi}
```

encoding 定义编码选项。初始化为 UTF8，使用时不能省略参数。

```
73 \DeclareStringOption[UTF8]{encoding}
```

pdffakebold 选择是否使用 PDF 原语生成伪粗体。默认是 `true`，如果选 `false` 则改用原来 CJK 宏包平移输出的伪粗体机制。

```
74 \DeclareBoolOption[true]{pdffakebold}
```

AutoFakeBold 使用 PDF 原语对 `b` 和 `bx` 系列生成伪粗体，并设置伪粗体描边线宽。初始化为 0.4 (bp)，无参数缺省值为 0.4。

```
75 \DeclareStringOption[0.4]{AutoFakeBold}[0.4]
```

AutoFakeSlant 使用 PDF 原语对 `s1` 形状生成伪斜体，并设置伪斜体倾斜度。初始化为 0 (禁用)，没有参数时缺省值为 0.167。

```
76 \DeclareStringOption[0]{AutoFakeSlant}[0.167]
```

执行选项。

```
77 \ProcessKeyvalOptions*
```

载入 CJK 相关支持包。

CJK: 基本 CJK 支持。

```
78 \RequirePackage{CJK}
```

`\zhm@check@encoding` 检查编码有效性。

```
79 \def\zhm@check@encoding{%
80   \unless\ifcsname CJK@\zhm@encoding @nfssenc\endcsname
81     \PackageError{zhmCJK}%
82     {There is no CJK encoding ‘\zhm@encoding’, please
83     check it over. ‘UTF8’ is used as default encoding.}%
84     {You can use ‘UTF8’, ‘GBK’, ‘Bg5’, ‘JIS’, ‘KS’, etc.
85     Read CJK.txt from CJK package for all valid encodings.}
86   \def\zhm@encoding{UTF8}
87   \fi}
```

检查全局编码。

```
88 \zhm@check@encoding
```

如果使用 UTF8 编码, 载入 CJKutf8 包。注意为避免与前面已经载入的 CJK 发生选项冲突, 在载入前模拟使用了 `encapsulated` 选项。

```
89 \begingroup
90 \def\zhm@UTF{UTF8}
91 \ifx\zhm@encoding\zhm@UTF\endgroup
92   \@namedef{opt@CJK.sty}{encapsulated}
93   \let\CJK@global\relax
94   \let\CJK@active\relax
95   \let\CJK@local\relax
96   \RequirePackage{CJKutf8}
97 \else\endgroup\fi
```

CJK 字符与西文字符之间的空格。

```
98 \RequirePackage{CJKspace}
```

CJK 标点禁则与压缩。

```
99 \RequirePackage{CJKpunct}
```

`\zhm@nfssenc` 保存当前编码对应的 NFSS 编码, 如 `\zhm@encoding` 为默认值 UTF8 时, 对应编码为 C70。

```
100 \edef\zhm@nfssenc{\@nameuse{CJK@\zhm@encoding @nfssenc}}
```

`\zhm@sfd` 保存当前编码对应的 SFD (subfont definition) 文件名, 不包括扩展名 `.sfd`。如 `\zhm@encodings` 为 UTF8 时, SFD 文件为 `Unicode.sfd`, 取值 `Unicode`。

```
101 \edef\zhm@sfd{\@nameuse{zhm@\zhm@encoding @sfd}}
```

下面定义已知的编码与 SFD 文件名对应。注意这里假定使用的都是 Unicode 内码的字体, SFD 文件名以 `U` 开头。常见 `TEX` 发行版附带的其他非 Unicode 的 SFD 文件不作考虑。

```
102 \def\zhm@def@sfd#1#2{\@namedef{zhm@#1@sfd}{#2}}
```

```
103 \zhm@def@sfd{UTF8}{Unicode}
```

```
104 \zhm@def@sfd{Bg5}{UBig5}
```

```
105 \zhm@def@sfd{Bg5+}{UBg5plus}
```

```
106 \zhm@def@sfd{JIS}{UJIS}
```

```
107 \zhm@def@sfd{KS}{UKS}
```

```
108 \zhm@def@sfd{GB}{UGB}
```

```
109 \zhm@def@sfd{GBK}{UGBK}
```

`\zhm@check@sfd` 检查编码对应的 SFD 文件是否支持。

```
110 \def\zhm@check@sfd{%
```

```
111   \unless\ifcsname zhm@\zhm@encoding @sfd\endcsname
```

```
112     \PackageError{zhmCJK}%
```

```
113       {No SFD file specified for ‘\zhm@encoding’ encoding.
```

```
114       Only ‘UTF8’, ‘Bg5’, ‘Bg5+’, ‘GB’, ‘GBK’, ‘JIS’ and
```

```
115       ‘KS’ are supported by now.}%
```

```
116     {}
```

```
117   \def\zhm@sfd{Unicode}
```

```
118   \fi}
```

```
119 \zhm@check@sfd
```

在导言区和正文中分别开启 CJK 的功能。非 UTF8 编码环境下, 导言区使用 `\CJK@makeActive` 设置 128 以上字符为活动字符, 并用 `\CJK@@enc` 定义这些活动字符。这里使用 `\CJK*` 代替原来的 CJK 环境, 可以避免在 `document` 环境内引入一层分组, 同时全局使用也不需要在全文最后使用 `\end{CJK}`, 最后也不需要 `\clearpage`。

```
120 \begingroup\def\zhm@UTF{UTF8}\expandafter\endgroup
```

```
121 \unless\ifx\zhm@encoding\zhm@UTF
```

```
122   \AtEndOfPackage{%
```

```
123     \CJK@makeActive
```

```
124     \CJK@@enc}
```

```
125 \fi
```

```

126 \AtBeginDocument{%
127   \@nameuse{CJK*}{\zhm@encoding}{\CJKfamilydefault}}

```

`\zhm@fontnum` 已定义的 CJK 字体数。

```
128 \newcount\zhm@fontnum
```

`\zhm@pdfliteral` 插入 PDF 原语 [3, 6]。用于实现伪斜体。

```

129 \ifpdf
130   \def\zhm@pdfliteral#1{\pdfliteral{#1}}
131 \else
132   \def\zhm@pdfliteral#1{\special{pdf:literal #1}}
133 \fi

```

`\zhm@pdfliteraldirect` 直接插入 PDF 原语 [3, 6]。用于实现伪粗体。相比 `\zhm@pdfliteral`，这里不用先保存 PDF 坐标等状态，生成的代码更直接。伪斜体采用坐标变换实现，因此不能不保存位置直接插入。

```

134 \ifpdf
135   \def\zhm@pdfliteraldirect#1{\pdfliteral direct {#1}}
136 \else
137   \def\zhm@pdfliteraldirect#1{\special{pdf:literal direct #1}}
138 \fi

```

使用 PDF 原语生成伪粗体，参考 [1, § 4.3.3, § 5.2.5]。这里伪粗体是通过将字体轮廓用 `AutoFakeBold` 参数指定的宽度进行描边得到的。

`\zhm@pdf@beginbold`

```

139 \def\zhm@pdf@beginbold{%
140   \zhm@pdfliteraldirect{2 Tr \@nameuse{zhm@\CJK@family @embolden} w}}

```

`\zhm@pdf@endbold`

```
141 \def\zhm@pdf@endbold{\zhm@pdfliteraldirect{0 Tr}}
```

`\ifzhm@fakebold` 判断当前字体是否使用伪粗体。当选项 `AutoFakeBold=0` 时，禁用伪粗体。

```

142 \def\zhm@zero{0}
143 \def\ifzhm@fakebold{\expandafter\unless\expandafter\ifx
144   \csname zhm@\zhm@fam @embolden\endcsname
145   \zhm@zero}
146 \def\zhm@fakeboldfalse{\@namedef{zhm@\zhm@fam @embolden}{0}}

```

使用 PDF 原语生成伪斜体，参考 [1, § 4.3.3]。这里伪斜体是通过局部坐标变换得到的，变换公式为：

$$\begin{aligned} \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} a & c \\ b & d \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\ &= \begin{pmatrix} 1 & \textit{slant} \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} x + \textit{slant} \cdot y \\ y \end{pmatrix} \end{aligned}$$

`\zhm@pdf@beginslant`

```
147 \def\zhm@pdf@beginslant{%
148   \zhm@pdfliteral{1 0 \@nameuse{zhm@CJK@family @slant} 1 0 0 cm}}
```

`\zhm@pdf@endslant`

```
149 \def\zhm@pdf@endslant{%
150   \zhm@pdfliteral{1 0 \@nameuse{zhm@CJK@family @negslant} 1 0 0 cm}}
```

`\ifzhm@CJK@bold@` `\ifzhm@CJK@bold@` 对应于 CJK 中的 `\ifCJK@bold@`，由 `\CJKbold` 和 `\CJKnormal` 控制，用于判断当前是否为伪粗体状态。

```
151 \newif\ifzhm@CJK@bold@
```

`\ifzhm@CJK@slant@` 类似 `\ifzhm@CJK@bold@`，由 `\CJKslant` 和 `\CJKnormal` 控制，用于判断当前是否为伪斜体状态。

```
152 \newif\ifzhm@CJK@slant@
```

`\CJKbold` 设置伪粗体（不斜）。根据 `pdffakebold` 选项选择伪粗体类型。

```
153 \ifzhm@pdffakebold
154   \def\CJKbold{\global\zhm@CJK@bold@true\global\zhm@CJK@slant@false}
155 \else
156   \def\CJKbold{\global\CJK@bold@true\global\zhm@CJK@slant@false}
157 \fi
```

`\CJKslant` 设置伪斜体（不加粗）。

```
158 \ifzhm@pdffakebold
159   \def\CJKslant{\global\zhm@CJK@bold@false\global\zhm@CJK@slant@true}
160 \else
161   \def\CJKnormal{\global\CJK@bold@false\global\zhm@CJK@slant@false}
162 \fi
```

`\CJKboldslant` 设置伪粗体加伪斜体。

```

163 \ifzhm@pdffakebold
164 \def\CJKboldslant{\global\zhm@CJK@bold@true\global\zhm@CJK@slant@true}
165 \else
166 \def\CJKnormal{\global\CJK@bold@false\global\zhm@CJK@slant@false}
167 \fi

```

`\CJKnormal` 恢复无伪粗、伪斜体的状态。

```

168 \ifzhm@pdffakebold
169 \def\CJKnormal{\global\zhm@CJK@bold@false\global\zhm@CJK@slant@false}
170 \else
171 \def\CJKnormal{\global\CJK@bold@false\global\zhm@CJK@slant@false}
172 \fi

```

`\ifzhm@fakeslant` 判断 `\zhm@fam` 确定的字体是否使用伪斜体。当选项 `AutoFakeSlant=0` 时，禁用伪斜体。

```

173 \def\ifzhm@fakeslant{\expandafter\unless\expandafter\ifx
174 \csname zhm@\zhm@fam @slant\endcsname
175 \zhm@zero}
176 \def\zhm@fakeslantfalse{%
177 \@namedef{zhm@\zhm@fam @slant}{0}%
178 \@namedef{zhm@\zhm@fam @negslant}{0}}

```

`\CJKsymbol` 重定义 `\CJKsymbol`，增加 PDF 的伪粗体、伪斜体功能。
注意原来的 `\CJKsymbol` 在 CJK 包中由 `\DeclareRobustCommand` 定义，因此这里只重定义 `\CJKsymbol` 后加空格的宏，修改其有效部分。这一定义不受 `\CJKpunct` 等宏包对 `\CJKsymbol` 附加重定义的影响。

```

179 \@namedef{CJKsymbol }#1{%
180 \ifzhm@CJK@slant@ \zhm@pdf@beginslant \fi
181 \ifzhm@CJK@bold@ \zhm@pdf@beginbold \fi
182 \symbol{#1}%
183 \ifCJK@bold@
184 \hbox to \CJKboldshift{\hss\symbol{#1}}%
185 \hbox to \CJKboldshift{\hss\symbol{#1}}%
186 \fi
187 \ifzhm@CJK@bold@ \zhm@pdf@endbold \fi
188 \ifzhm@CJK@slant@ \zhm@pdf@endslant \fi}

```

`\zhm@DeclareFontShape` `\DeclareFontShape` 内部需要修改 `\catcode`，因此在其他宏中声明时，使用 `\scantokens` 处理。

```

189 \def\zhm@DeclareFontShape#1#2#3#4#5#6{%
190   \scantokens{\DeclareFontShape{#1}{#2}{#3}{#4}{#5}{#6}}}
```

类似 CJKb，定义 CJKsl 和 CJKbsl 尺寸函数，用于伪斜体、伪粗斜体的字形声明。

```

191 \DeclareSizeFunction{CJKsl}{\CJK@empty\@font@warning}
192 \DeclareSizeFunction{CJKbsl}{\CJK@empty\@font@warning}
```

`\zhm@mapline` 添加实际字体映射行，两个参数是字体编号和字体文件名。为 pdfTeX 与 DVIPDFMx 引擎使用不同的命令完成 [6]。

```

193 \ifpdf
194   \def\zhm@mapline#1#2{%
195     \pdfmapline{=zhm#1@\zhm@sfd @ <#2}}
196 \else
197   \def\zhm@mapline#1#2{%
198     \special{pdf:mapline zhm#1@\zhm@sfd @ \zhm@cmap\space \ifzhm@embed\else!\fi#2}}
199 \fi
```

`\zhm@setfontmap` 设置字体映射，两个参数是字体编号和字体文件名。字体映射需要在输出例程初始处设置。同时将映射行完全展开后保存起来备用。

`\zhm@maplines`

```

200 \def\zhm@setfontmap#1#2{%
201   \AtBeginDvi{\zhm@mapline{#1}{#2}}%
202   \xdef\zhm@maplines{%
203     \unexpanded\expandafter{\zhm@maplines}%
204     \zhm@mapline{#1}{#2}}}
```

如果载入了 atbegshi 宏包，则还要处理修改了的输出例程。这会影响到 eso-pic 等用户层宏包。

```

205 \AtBeginDocument{%
206   \@ifpackageloaded{atbegshi}{%
207     \AtBeginShipoutFirst{\zhm@maplines}}{}}
208 \let\zhm@maplines\empty
```

`\zhm@DeclareRealFont` 分配实际字体，并设置字体映射。参数是字体文件名。如果实际字体已经分配，则不改变。

```

209 \def\zhm@DeclareRealFont#1{
210   \unless\ifcsname zhm#1@fontnum\endcsname
211     \global\advance\zhm@fontnum\@ne
212     \ifnum\zhm@fontnum>\@xxxii
213       \PackageError{zhmCJK}%
214         {No more CJK font families can be setup.}%
```



```

215     {There are at most 32 families setup by zhmCJK.}
216   \else
217     \expandafter\xdef\csname zhm@#1@fontnum\endcsname{\the\zhm@fontnum}
218     \zhm@setfontmap{\the\zhm@fontnum}{#1}
219   \fi
220 \fi
221 \edef\zhm@tfm{zhm\@nameuse{zhm@#1@fontnum}}

222 \define@key{zhm}{cmap}{%
223   \edef\zhm@cmap{#1}}

224 \define@key{zhm}{embed}[true]{%
225   \ifpdf
226     \PackageWarning{zhmCJK}
227     {The ‘embed’ option is not supported by pdfTeX driver.\MessageBreak
228     Ignoring option ‘embed’}
229   \fi
230   \edef\zhm@embed{#1}%
231   \ifx\zhm@embed\zhm@true
232     \zhm@embedtrue
233   \else\ifx\zhm@embed\zhm@false
234     \zhm@embedfalse
235   \else
236     \PackageError{zhmCJK}%
237     {The ‘embed’ option must be true or false.}%
238     {See the package manual for more information.}%
239   \fi\fi}
240 \def\zhm@true{true}
241 \def\zhm@false{false}

242 \define@key{zhm}{encoding}{%
243   \edef\zhm@encoding{#1}%
244   \zhm@check@encoding
245   \edef\zhm@nfssenc{\@nameuse{CJK@\zhm@encoding @nfssenc}}%
246   \edef\zhm@sfd{\@nameuse{zhm@\zhm@encoding @sfd}}%
247   \zhm@check@sfd}

    设置字体 AutoFakeBold 选项。缺省值是非 0 的全局选项值，或 0.4。
248 \define@key{zhm}{AutoFakeBold}{%
249   [\ifx\zhm@AutoFakeBold\zhm@zero 0.4\else\zhm@AutoFakeBold \fi]%
250   {\expandafter\xdef\csname zhm@\zhm@fam @embolden\endcsname{#1}}

```

`\zhm@neg` 利用长度计算取负数。

```
251 \def\zhm@neg#1{\strip@pt\dimexpr-#1pt\relax}
```

设置字体 `AutoFakeSlant` 选项。缺省值是非 0 的全局选项值，或 0.167。

```
252 \define@key{zhm}{AutoFakeSlant}{%
253   [\ifx\zhm@AutoFakeSlant\zhm@zero 0.167\else\zhm@AutoFakeSlant \fi]%
254   {\expandafter\xdef\csname zhm@\zhm@fam @slant\endcsname{#1}%
255   \expandafter\xdef\csname zhm@\zhm@fam @negslant\endcsname{\zhm@neg{#1}}}
```

设置 `BoldFont` 选项。

```
256 \define@key{zhm}{BoldFont}{
257   \zhm@DeclareRealFont{#1}
258   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{n}{<-> CJK * \zhm@tfm}%
259   {\CJKnormal}
260   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{n}{<-> CJK * \zhm@tfm}%
261   {\CJKnormal}
262   \zhm@fakeboldfalse}
```

设置 `ItalicFont` 选项。

```
263 \define@key{zhm}{ItalicFont}{
264   \zhm@DeclareRealFont{#1}
265   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{it}{<-> CJK * \zhm@tfm}%
266   {\CJKnormal}}
```

设置 `BoldItalicFont` 选项。

```
267 \define@key{zhm}{BoldItalicFont}{
268   \zhm@DeclareRealFont{#1}
269   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{it}{<-> CJK * \zhm@tfm}%
270   {\CJKnormal}
271   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{it}{<-> CJK * \zhm@tfm}%
272   {\CJKnormal}}
```

设置 `SlantedFont` 选项。

```
273 \define@key{zhm}{SlantedFont}{
274   \zhm@DeclareRealFont{#1}
275   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{sl}{<-> CJK * \zhm@tfm}%
276   {\CJKnormal}
277   \zhm@fakeslantfalse}
```

设置 `BoldSlantedFont` 选项。

```
278 \define@key{zhm}{BoldSlantedFont}{
279   \zhm@DeclareRealFont{#1}
280   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{sl}{<-> CJK * \zhm@tfm}%
```

```

281     {\CJKnormal}
282     \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{sl}{<-> CJK * \zhm@tfm}%
283     {\CJKnormal}}

```

`\zhm@newfontfamily` 设置一个 CJK 新字体族。三个参数分别为：字体族名，选项，字体文件名。

```

284 \def\zhm@newfontfamily#1[#2]#3{
285     \begingroup

```

警告字体族重定义

```

286     \ifcsname zhm@#1@font\endcsname
287     \PackageWarning{zhmCJK}{Redefinition of CJK family ‘#1’}
288     \fi

```

缓存字体族名及对应主字体文件名。

```

289     \edef\zhm@fam{#1}
290     \expandafter\xdef\csname zhm@#1@font\endcsname{#3}

```

使全局的 `AutoFakeBold` 和 `AutoFakeSlant` 生效。

```

291     \setkeys{zhm}{%
292         AutoFakeBold=\zhm@AutoFakeBold,
293         AutoFakeSlant=\zhm@AutoFakeSlant}

```

定义 CJK 字体族。

```

294     \DeclareFontFamily{\zhm@nfssenc}{\zhm@fam}{\hyphenchar\font\m@ne}

```

执行选项，定义编码、CMap 和其他字形等。

```

295     \setkeys{zhm}{#2}

```

检查并定义实际字体，定义标准字形。模拟 `.fd` 文件的定义 [5]。

```

296     \zhm@DeclareRealFont{#3}
297     \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{n}{<-> CJK * \zhm@tfm}%
298     {\CJKnormal}

```

按需要，定义伪粗体、伪斜体字形。

```

299     \ifzhm@fakebold
300         \zhm@DeclareRealFont{#3}
301         \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{n}%
302             {<-> CJKb * \zhm@tfm}{\CJKbold}
303         \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{n}%
304             {<-> CJKb * \zhm@tfm}{\CJKbold}
305     \fi
306     \ifzhm@fakeslant
307         \zhm@DeclareRealFont{#3}

```

```

308 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{sl}%
309 {<-> CJKsl * \zhm@tfm}{\CJKslant}
310 \fi
311 \expandafter\ifzhm@fakeslant\ifzhm@fakebold
312 \zhm@DeclareRealFont{#3}
313 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{sl}%
314 {<-> CJKbsl * \zhm@tfm}{\CJKboldslant}
315 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{sl}%
316 {<-> CJKbsl * \zhm@tfm}{\CJKboldslant}
317 \fi\fi
318 \endgroup}

```

`\zhm@newfontfamily@post` 为了支持字体属性可选项在前在后两种语法，给出两个辅助工具。

```

\zhm@newfontfamily@postaux 319 \def\zhm@newfontfamily@post#1#2{%
320 \@ifnextchar [%]
321 {\zhm@newfontfamily@postaux{#1}{#2}}%
322 {\zhm@newfontfamily{#1}[] {#2}}
323 \def\zhm@newfontfamily@postaux#1#2[#3]{%
324 \zhm@newfontfamily{#1} [{#3}] {#2}}

```

`\setCJKfamilyfont` 设置一个 CJK 字体族。实际有三个参数，语法与 `\zhm@newfontfamily` 大致相同，第二个参数可选。

```

325 \newcommand*\setCJKfamilyfont[1]{%
326 \@ifnextchar [%]
327 {\zhm@newfontfamily{#1}}%
328 {\zhm@newfontfamily@post{#1}}
329 \@onlypreamble\setCJKfamilyfont

```

`\setCJKmainfont` 设置 CJK 普通（罗马）字体。

```

330 \newcommand*\setCJKmainfont{%
331 \setCJKfamilyfont{\CJKrmdefault}}
332 \@onlypreamble\setCJKmainfont

```

`\setCJKromanfont` `\setCJKmainfont` 的别名。

```

333 \let\setCJKromanfont\setCJKmainfont
334 \@onlypreamble\setCJKromanfont

```

`\setCJKsansfont` 设置 CJK 无衬线字体。

```

335 \newcommand*\setCJKsansfont{%
336 \setCJKfamilyfont{\CJKsfdefault}}
337 \@onlypreamble\setCJKsansfont

```

```

\setCJKmonofont 设置 CJK 等宽（打字机）字体。
338 \newcommand*\setCJKmonofont{%
339   \setCJKfamilyfont{\CJKttdefault}}
340 \@onlypreamble\setCJKmonofont

\CJKrmdefault CJK 罗马体默认字体族，作用于 \rmfamily。
341 \providecommand*\CJKrmdefault{rm}

\CJKsfdefault CJK 无衬线体默认字体族，作用于 \sffamily。
342 \providecommand*\CJKsfdefault{sf}

\CJKttdefault CJK 打字机体默认字体族，作用于 \ttfamily。
343 \providecommand*\CJKttdefault{tt}

\CJKfamilydefault CJK 默认字体族，作用于 \normalfont。
344 \providecommand*\CJKfamilydefault{\CJKrmdefault}

```

重定义 \normalfont, \rmfamily, \sffamily 和 \ttfamily, 使其同时设置 CJK 字体。

```

345 \DeclareRobustCommand\normalfont
346     {\CJKfamily{\CJKfamilydefault}%
347     \usefont\encodingdefault
348         \familydefault
349         \seriesdefault
350         \shapedefault
351     \relax}
352 \let\reset@font\normalfont
353 \DeclareRobustCommand\rmfamily
354     {\not@math@alphabet\rmfamily\mathrm
355     \fontfamily\rmdefault\CJKfamily{\CJKrmdefault}\selectfont}
356 \DeclareRobustCommand\sffamily
357     {\not@math@alphabet\sffamily\mathsf
358     \fontfamily\sfddefault\CJKfamily{\CJKsfdefault}\selectfont}
359 \DeclareRobustCommand\ttfamily
360     {\not@math@alphabet\ttfamily\mathtt
361     \fontfamily\ttdefault\CJKfamily{\CJKttdefault}\selectfont}

```

\zhm@setup@pinyin@UTF8 下面简单处理 UTF-8 编码下的汉语拼音符号，让下列拼音符号

```

āáǎà ōóòò ēéěè ê īíì ùúùù ūǔǔù ùǔǔù ñññ
ĀĀĀĀ ŌŌŌŌ ĒĒĒĒ Ê Î Î Î Û Û Û Û Ū Ū Ū Ū Û Û Û Û Ñ Ñ Ñ

```

可以直接输入，不受 CJK 包影响。

```
362 \@namedef{zhm@setup@pinyin@UTF8}{%
363   \@namedef{u8:\detokenize{ā}}{\={a}}%
364   \@namedef{u8:\detokenize{á}}{\'a}%
365   \@namedef{u8:\detokenize{ǎ}}{\v{a}}%
366   \@namedef{u8:\detokenize{â}}{\'a}%
367   \@namedef{u8:\detokenize{ō}}{\={o}}%
368   \@namedef{u8:\detokenize{ó}}{\'o}%
369   \@namedef{u8:\detokenize{ö}}{\v{o}}%
370   \@namedef{u8:\detokenize{ò}}{\'o}%
371   \@namedef{u8:\detokenize{ē}}{\={e}}%
372   \@namedef{u8:\detokenize{é}}{\'e}%
373   \@namedef{u8:\detokenize{ě}}{\v{e}}%
374   \@namedef{u8:\detokenize{ê}}{\'e}%
375   \@namedef{u8:\detokenize{ê}}{\^e}%
376   \@namedef{u8:\detokenize{ī}}{\={i}}%
377   \@namedef{u8:\detokenize{í}}{\'i}%
378   \@namedef{u8:\detokenize{i}}{\v{i}}%
379   \@namedef{u8:\detokenize{i}}{\'i}%
380   \@namedef{u8:\detokenize{ū}}{\={u}}%
381   \@namedef{u8:\detokenize{ú}}{\'u}%
382   \@namedef{u8:\detokenize{ü}}{\v{u}}%
383   \@namedef{u8:\detokenize{ü}}{\'u}%
384   \@namedef{u8:\detokenize{ū}}{\={u}}%
385   \@namedef{u8:\detokenize{ú}}{\'u}}%
386   \@namedef{u8:\detokenize{ü}}{\v{u}}%
387   \@namedef{u8:\detokenize{ü}}{\'u}}%
388   \@namedef{u8:\detokenize{ü}}{\=u}}%
389   \@namedef{u8:\detokenize{ñ}}{\'n}}%
390   \@namedef{u8:\detokenize{ñ}}{\v{n}}%
391   \@namedef{u8:\detokenize{n}}{\'n}}%
392   \@namedef{u8:\detokenize{Ā}}{\={A}}%
393   \@namedef{u8:\detokenize{Á}}{\'A}}%
394   \@namedef{u8:\detokenize{Ā}}{\v{A}}%
395   \@namedef{u8:\detokenize{Ā}}{\'A}}%
396   \@namedef{u8:\detokenize{Ō}}{\={O}}%
397   \@namedef{u8:\detokenize{Ō}}{\'O}}%
398   \@namedef{u8:\detokenize{Ō}}{\v{O}}%
399   \@namedef{u8:\detokenize{Ō}}{\'O}}%
400   \@namedef{u8:\detokenize{Ē}}{\={E}}%
401   \@namedef{u8:\detokenize{Ē}}{\'E}}%
```

```

402 \@namedef{u8:\detokenize{Ě}}{\v{E}}%
403 \@namedef{u8:\detokenize{Ě}}{\‘{E}}%
404 \@namedef{u8:\detokenize{Ê}}{\˘{E}}%
405 \@namedef{u8:\detokenize{Ñ}}{\’{N}}%
406 \@namedef{u8:\detokenize{Ñ}}{\v{N}}%
407 \@namedef{u8:\detokenize{Ñ}}{\‘{N}}%
408 }

```

`\zhm@setup@pinyin` 调用具体编码下的拼音设置。目前只有 UTF-8 编码有合理的设置，其他编码较难实现。

```

409 \def\zhm@setup@pinyin{%
410   \@nameuse{zhm@setup@pinyin@\zhm@encoding}}
411 \zhm@setup@pinyin
412 </package>

```

5.2 TFM 及映射文件生成脚本 zhmCJK.lua

这里字体脚本的实现参考了 zhmetrics 项目中吴凌云¹、zwhuang² 编写的 zhtfm.lua 脚本。其中 TFM 生成原理参见 [8]。

```

1  -- Copyright (C) 2012--2016 by Leo Liu <leoliu.pku@gmail.com>
2  -----
3  -- This work may be distributed and/or modified under the
4  -- conditions of the LaTeX Project Public License, either version 1.3
5  -- of this license or (at your option) any later version.
6  -- The latest version of this license is in
7  --   http://www.latex-project.org/lppl.txt
8  -- and version 1.3 or later is part of all distributions of LaTeX
9  -- version 2005/12/01 or later.
10 --
11 -- This work has the LPPL maintenance status ‘maintained’.
12 --
13 -- The Current Maintainer of this work is Leo Liu.
14 --
15 -- This work consists of the files zhmCJK.dtx,
16 --                               zhmCJK.ins,
17 --                               zhmCJK.lua,
18 --                               zhmCJK-addmap.pl,
19 --                               and the derived files zhmCJK.sty,

```

¹中科院应用数学研究所副研究员。wulingyun@gmail.com

²zwhuang@gmail.com

```

20 --          zhmCJK.pdf,
21 --          zhmCJK-test.tex,
22 --          README,                      (from zhmCJK.dtx)
23 --          zhmCJK.map,
24 --          texfonts.map.template,
25 --          zhmCJK.tfm,
26 --          miktex-tfm/zhm*/zhm**.tfm. (from zhmCJK.lua)
27 --
28 -----
29 -- This lua script is used to generate zhm*.tfm and zhmCJK.map.
30 --
31 -- Usage:
32 --
33 --     texlua zhmCJK.lua map
34 --
35 -- or (for MiKTeX only)
36 --
37 --     texlua zhmCJK.lua nomap
38 --
39
40 -----
41 -- OS dependent settings
42 -----
43
44 if os.type == "unix" then
45     path_slash = "/"
46 else
47     path_slash = "\\"
48 end
49
50 -----
51 -- Generate TFM
52 -----
53
54 pl_template = {
55     char = [[
56 (CHARACTER D %d
57     (CHARWD R 1.0)
58     (CHARHT R 0.8)
59     (CHARDP R 0.1)
60     )

```



```

61  ]],
62      header = [[
63      (FAMILY %s)
64      (CODINGScheme CJK-UNICODE)
65      (DESIGNSIZE R 10.0)
66      (HEADER D 18 H 43726561)
67      (HEADER D 19 H 74656420)
68      (HEADER D 20 H 62792060)
69      (HEADER D 21 H 7A686D43)
70      (HEADER D 22 H 4A4B2E6C)
71      (HEADER D 23 H 7561272E)
72      (HEADER D 24 H 20436F70)
73      (HEADER D 25 H 79726967)
74      (HEADER D 26 H 68742028)
75      (HEADER D 27 H 43292032)
76      (HEADER D 28 H 30313220)
77      (HEADER D 29 H 6279204C)
78      (HEADER D 30 H 656F204C)
79      (HEADER D 31 H 6975203C)
80      (HEADER D 32 H 6C656F6C)
81      (HEADER D 33 H 69752E70)
82      (HEADER D 34 H 6B754067)
83      (HEADER D 35 H 6D61696C)
84      (HEADER D 36 H 2E636F6D)
85      (HEADER D 37 H 3E0D0A00)
86      (CHECKSUM 0 0)
87      (FONTDIMEN
88          (SLANT R %f)
89          (SPACE R 0.5)
90          (STRETCH R 0.3)
91          (SHRINK R 0.1)
92          (XHEIGHT R 0.4)
93          (QUAD R 1.0)
94      )
95  ]],
96  -- charset = (defined below)
97  }
98
99  function pl_charset()
100      local charset = {}
101      for cid = 0, 0xff do

```

```

102         charset[#charset + 1] = string.format(pl_template.char, cid)
103     end
104     return table.concat(charset)
105 end
106
107 pl_template.charset = pl_charset()
108
109 function write_tfm(path, name, slant)
110     local fullname = path .. path_slash .. name
111     local s_pl = string.format(pl_template.header, string.upper(name), slant)
112         .. pl_template.charset
113     local f_pl = io.open(fullname .. ".pl", "w")
114     f_pl:write(s_pl)
115     f_pl:close()
116     os.execute([[pltotf ]] .. fullname .. [[.pl" ]] .. fullname .. [[.tfm]])
117     os.remove(fullname .. ".pl")
118 end
119
120 -----
121 -- Main functions to generate necessary files
122 -----
123
124 -- For TeX Live and other TeX distributions where texfonts.map is supported,
125 -- we generate texfonts.map.template, zhmCJK.map, and zhmCJK.tfm.
126 function generate_with_fontmap()
127     local f_map = io.open("zhmCJK.map", "w")
128     for fam = 1, 32 do
129         for sid = 0, 0xff do
130             f_map:write(string.format("zhmCJK.tfm zhm%d%02x.tfm\n",
131                 fam, sid))
132         end
133     end
134     f_map:write("\n")
135     f_map:close()
136
137     f_map = io.open("texfonts.map.template", "w")
138     f_map:write("include zhmCJK.map\n\n")
139     f_map:close()
140
141     write_tfm(".", "zhmCJK", 0.0)
142 end

```

```

143
144 -- For MiKTeX and other TeX distributions where texfonts.map is not supported,
145 -- we generate a lot of zhmXYX.tfm, where X from 1 to 32, Y from 0x00 to 0xff.
146 function generate_without_fontmap()
147     lfs.mkdir("miktex-tfm")
148     for fam = 1, 32 do
149         local path = string.format("miktex-tfm" .. path_slash .. "zhm%d", fam)
150         lfs.mkdir(path)
151         print(path)
152         for sid = 0, 0xff do
153             local name = string.format("zhm%d%02x", fam, sid)
154             write_tfm(path, name, 0.0)
155         end
156     end
157 end
158
159 -----
160 -- Console User Interface
161 -----
162
163 help_info = [[
164 Usage:
165
166     texlua ]].. arg[0] .. [[ map|nomap
167
168     map:    Generate a public TFM shared by all CJK fonts with mapping files.
169             It is suggested for TeX Live and other web2c distributions.
170
171     nomap:  Generate all TFM files for CJK fonts into "miktex-tfm" directory.
172             MiKTeX needs this since it does not support TFM mappings.
173 ]]
174
175 if #arg ~= 1 then
176     print(help_info)
177 else
178     if arg[1] == "map" then
179         generate_with_fontmap()
180     elseif arg[1] == "nomap" then
181         generate_without_fontmap()
182     else
183         print("! Unknown option " .. arg[1])

```

```

184         print(help_info)
185     end
186 end
187
188 -- end of file zhmCJK.lua --
189 --
190 -----
191 --
192 -- The code is inspired by zhtfm.lua of zhmetrics bundle.
193 --
194 -- The zhmetrics bundle is distributed under LPPL 1.3 or later. The original
195 -- authors of zhtfm.lua are:
196 --
197 --         Lingyun Wu    <wulingyun@gmail.com>
198 --         zwhuang      <zwhuang@gmail.com>
199 --
200 -- For more information of zhmetrics, see the ctex-kit project:
201 --
202 --         https://github.com/CTeX-org/ctex-kit
203 --
204

```

6 版本历史

v0.1	项。	17
General: 初始版本	设置字体 <code>BoldFont</code> 选项。 . .	18
v0.2	设置字体 <code>BoldItalicFont</code> 选	
General: 编写宏包文档。增加	项。	18
<code>CJKpunct</code> 。做一些小的代码	设置字体 <code>BoldSlantedFont</code>	
调整。	选项。	18
v0.3	设置字体 <code>ItalicFont</code> 选项。	18
<code>\setCJKromanfont</code> : 新增。 . .	设置字体 <code>SlantedFont</code> 选项。	
General: 使用 PDF 原语生成伪	18
粗体		13
增加宏包 <code>pdffakebold</code> 选项		10
v0.4		
General: 增加宏包		
<code>AutoFakeBold</code> 选项		10
设置字体 <code>AutoFakeBold</code> 选		
	v0.5	
	General: 使用 Lua 脚本生成	
	TFM 和映射文件; 为	
	MiKTeX 生成单独的 TFM	
	文件。将原来的	
	<code>zhmetrics.tfm</code> 改名为	

zhmCJK.tfm, 不再依赖原有的 zhmetrics 包。	1	体。	10
v0.6		支持伪粗斜体。	16
General: 增加字体		设置字体 cmap 选项。	17
AutoFakeSlant 选项	18	设置字体 embed 选项。	17
增加宏包 AutoFakeSlant 选项	10	重新实现伪粗体、伪斜体功能, 使之可以正常处理断行、标点压缩等问题。	15
v0.7		v0.9a	
\setCJKfamilyfont: 修改可选参数位置, 与 xeCJK 包语法一致。	20	\zhm@maplines: 保存重复的字体映射。(李清)	16
\zhm@newfontfamily: 更改语法, 修改选项位置。	19	\zhm@setup@pinyin@UTF8: 处理 UTF-8 编码下的拼音符号输入。	21
General: 宏包载入时进行编译引擎测试	9	General: 使用 CJK* 的环境头代替完整 CJK* 环境, 可以减少一个全局分组。(由李清建议)	12
生成 TFM 时加入版权说明。	28	v0.9b	
v0.8		General: UTF8 编码不使用 \CJK@makeActive。	12
General: 增加宏包 encoding 选项选择默认编码。	10	使用 \CJK@@enc 避免一些定义问题。	12
设置字体 encoding 选项。	17	v0.9c	
v0.9		General: 支持字体属性可选项在后的新语法。	20
General: 增加宏包 cmap 选项选择 DVIPDFMx 的 CMap 映射。	10		
增加宏包 embed 选项, 允许 DVIPDFMx 驱动不嵌入字			

7 代码索引

斜体的数字表示对应项说明所在的页码。下划线的数字表示定义所在的代码行号; 而直立体数字表示对应项使用时所在的行号。

Symbols	384, 392, 396, 400	AutoFakeBold (option)	<u>75</u>
\"	384,	\^	375, 404
	385, 386, 387, 388	\‘	366, 370, 374, 379,
\’	364, 368, 372, 377,		383, 387, 391,
	381, 385, 389,		395, 399, 403, 407
	393, 397, 401, 405	A	
\=	363, 367,	\AddToKeyvalOption	67
	371, 376, 380,		
		C	
		\CJK@@enc	124
		\CJK@active	94
		\CJK@bold@false	

...	161, 166, 171	377, 378, 379,	
\CJK@bold@true ..	156	380, 381, 382,	P
\CJK@empty ..	191, 192	383, 384, 385,	pdffakebold (op-
\CJK@family		386, 387, 388,	tion) 74
...	140, 148, 150	389, 390, 391,	\pdfliteral . 130 , 135
\CJK@global	93	392, 393, 394,	\pdfmapline 195
\CJK@local	95	395, 396, 397,	R
\CJK@makeActive .	123	398, 399, 400,	\reset@font 352
\CJKbold 153 , 302 , 304		401, 402, 403,	\rmdefault
\CJKboldshift 184 , 185		404, 405, 406, 407	\rmfamily ... 353 , 354
\CJKboldslant			
...	163 , 314 , 316		S
\CJKfamily ... 26 ,		embed (option) 66	\selectfont
346 , 355 , 358 , 361		\empty 208	... 355 , 358 , 361
\CJKfamilydefault		encoding (option) . 73	\seriesdefault .. 349
.. 4 , 127 , 344 , 346		\encodingdefault 347	\setCJKfamilyfont
\CJKnormal 161 , 166 ,		F 3 , 10 ,
168 , 259 , 261 ,		\familydefault .. 348	325 , 331 , 336 , 339
266 , 270 , 272 ,		\fontfamily	\setCJKmainfont ..
276 , 281 , 283 , 298		... 355 , 358 , 361	... 2 , 7 , 330 , 333
\CJKrmdefault .. 4 ,		I	\setCJKmonofont ..
331 , 341 , 344 , 355		\i ... 376 , 377 , 378 , 379 3 , 9 , 338
\CJKsfdefault		\ifCJK@bold@ ... 183	\setCJKromanfont .
.. 4 , 336 , 342 , 358		\ifzhm@CJK@bold@ 3 , 333
\CJKslant ... 158 , 309		... 151 , 181 , 187	\setCJKsansfont ..
\CJKsymbol	179	\ifzhm@CJK@slant@ 3 , 8 , 335
\CJKttdefault 152 , 180 , 188	\sfdefault
.. 4 , 339 , 343 , 361		\ifzhm@embed ... 198	\sffamily ... 356 , 357
cmap (option)	65	\ifzhm@fakebold ..	\shapedefault ... 350
		... 142 , 299 , 311	\special 132 , 137 , 198
D		\ifzhm@fakeslant .	T
\DeclareFontFamily		... 173 , 306 , 311	\ttdefault
..... 294		\ifzhm@pdffakebold	... 359 , 360
\DeclareFontShape 190		153 , 158 , 163 , 168	
\DeclareSizeFunction		\ifzhm@primitive .	U
..... 191 , 192		... 31 , 36 , 45 , 52	\usefont
\detokenize		M	347
..... 363 , 364 ,		\MessageBreak 70 , 227	V
365 , 366 , 367 ,			\v 365 , 369 , 373 , 378 ,
368 , 369 , 370 ,		N	382 , 386 , 390 ,
371 , 372 , 373 ,		\normalfont . 345 , 352	394 , 398 , 402 , 406
374 , 375 , 376 ,			

Z		
<code>\zhm@AutoFakeBold</code>	<code>\zhm@embedfalse</code> . 234	245, 258, 260,
..... 249, 292	<code>\zhm@embedtrue</code> .. 232	265, 269, 271,
<code>\zhm@AutoFakeSlant</code>	<code>\zhm@encoding</code>	275, 280, 282,
..... 253, 293	80, 82, 86, 91,	294, 297, 301,
<code>\zhm@check@encoding</code>	100, 101, 111,	303, 308, 313, 315
..... 79, 88, 244	113, 121, 127,	<code>\zhm@pdf@beginbold</code>
<code>\zhm@check@sfd</code> ...	243, 245, 246, 410 139, 181
..... 110, 247	<code>\zhm@fakeboldfalse</code>	<code>\zhm@pdf@beginslant</code>
<code>\zhm@CJK@bold@false</code> 146, 262 147, 180
..... 159, 169	<code>\zhm@fakeslantfalse</code>	<code>\zhm@pdf@endbold</code> .
<code>\zhm@CJK@bold@true</code> 176, 277 141, 187
..... 154, 164	<code>\zhm@false</code> .. 233, 241	<code>\zhm@pdf@endslant</code>
<code>\zhm@CJK@slant@false</code>	<code>\zhm@fam</code> . 144, 146, 149, 188
.... 154, 156,	174, 177, 178,	<code>\zhm@pdfliteral</code> ..
161, 166, 169, 171	250, 254, 255,	... 129, 148, 150
<code>\zhm@CJK@slant@true</code>	258, 260, 265,	<code>\zhm@pdfliteraldirect</code>
..... 159, 164	269, 271, 275,	... 134, 140, 141
<code>\zhm@cmmap</code> ... 198, 223	280, 282, 289,	<code>\zhm@setfontmap</code> ..
<code>\zhm@DeclareFontShape</code>	294, 297, 301, 200, 218
.... 189, 258,	303, 308, 313, 315	<code>\zhm@setup@pinyin</code> 409
260, 265, 269,	<code>\zhm@fontnum</code> 128,	<code>\zhm@setup@pinyin@UTF8</code>
271, 275, 280,	211, 212, 217, 218 362
282, 297, 301,	<code>\zhm@mapline</code>	<code>\zhm@sfd</code> 101,
303, 308, 313, 315	... 193, 201, 204	117, 195, 198, 246
<code>\zhm@DeclareRealFont</code>	<code>\zhm@maplines</code>	<code>\zhm@tfm</code> . 221, 258,
209, 257, 264,	... 200, 207, 208	260, 265, 269,
268, 274, 279,	<code>\zhm@neg</code> 251, 255	271, 275, 280,
296, 300, 307, 312	<code>\zhm@newfontfamily</code>	282, 297, 302,
<code>\zhm@def@sfd</code> 102,	284, 322, 324, 327	304, 309, 314, 316
103, 104, 105,	<code>\zhm@newfontfamily@post</code>	<code>\zhm@true</code> ... 231, 240
106, 107, 108, 109 319, 328	<code>\zhm@UTF</code>
<code>\zhm@embed</code>	<code>\zhm@newfontfamily@postaux</code> . 90, 91, 120, 121	<code>\zhm@zero</code> ... 142,
... 230, 231, 233 319	<code>\zhm@nfssenc</code> 100,
		145, 175, 249, 253

8 参考文档

- [1] ADOBE SYSTEMS INCORPORATED. *PDF reference: Adobe portable document format version 1.4*. Addison-Wesley, third edition, 2001
http://www.adobe.com/devnet/pdf/pdf_reference.html

- [2] KARL BERRY, OLAF WEBER AND TACO HOEKWATER. *Kpathsea library*, version 6.0.1, 2011
[CTAN://info/doc-k/kpathsea.pdf](http://ctan.org/info/doc-k/kpathsea.pdf)
- [3] JIN-HWAN CHO. “DVI specials for PDF generation”. *TUGboat*, volume 30(1):pages 6–11, 2009
<http://www.tug.org/TUGboat/Articles/tb30-1/tb94cho.pdf>
- [4] WERNER LEMBERG. *CJK*, version 4.8.2, 2008
[CTAN://language/chinese/CJK/cjk-4.8.2/doc/CJK.txt](http://ctan.org/language/chinese/CJK/cjk-4.8.2/doc/CJK.txt)
- [5] WERNER LEMBERG. *The fonts*, 2008
[CTAN://language/chinese/CJK/cjk-4.8.2/doc/fonts.txt](http://ctan.org/language/chinese/CJK/cjk-4.8.2/doc/fonts.txt)
- [6] HÀN THẾ THÀNH. “A closer look at TrueType fonts and pdf \TeX ”. *TUGboat*, volume 30(1):pages 32–34, 2009
<http://www.tug.org/TUGboat/Articles/tb30-1/tb94thanh.pdf>
- [7] 孙文昌. *xeCJK* 宏包, 版本 2.4.5, 2012
[CTAN://macros/xetex/latex/xecjk/xeCJK.pdf](http://ctan.org/macros/xetex/latex/xecjk/xeCJK.pdf)
- [8] 王越. *CJK* 宏包中, 中文字体的秘密. CTEX 论坛, 2009 年 5 月
<http://bbs.ctex.org/viewthread.php?tid=50078>