

zhmCJK 宏包

刘海洋

leoliu.pku@gmail.com

2013/08/10 v0.9

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 pdf \TeX 和 DVIPDFMx 两种输出驱动，可以使用 pdf latex 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 <code>unicode</code> , 相当于使用基于 UCS2 的字体编码; 部分字体需要设置为 <code>UniGB-UTF16-H</code> 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[⟨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[BoldFont=simhei.ttf,ItalicFont=simkai.ttf]{simsun.ttc}
```

将使用文件名为 `simsun.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
  [cmap=UniGB-UTF16-H,
   BoldFont=FandolSong-Bold.otf,
   ItalicFont=FandolKai-Regular.otf]
{FandolSong-Regular.otf}
```

可以设置 `Fandol` 字体作为正文字体。

`\setCJKsansfont` `\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` [(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>} [(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}[BoldFont=msyhbd.ttf]{msyh.ttf}
```

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.ttf`（微软雅黑），粗体形式是 `msykbd.ttf`（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[BoldFont=simhei.ttf,ItalicFont=simkai.ttf]{simsun.ttc}
8 \setCJKsansfont[AutoFakeBold=0]{simhei.ttf}
9 \setCJKmonofont[AutoFakeSlant]{simfang.ttf}
10 \setCJKfamilyfont{xinwei}{STXINWEI.TTF}
11
12 \title{\bfseries 一个华丽的 zhmCJK 测试}
13 \author{Leo Liu}
14 \date{2012 年 2 月 6 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。

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 -T_EX 支持, 并要求使用较新版本的 pdfT_EX 或 DVIPDFM_x 生成 PDF 文件。

3.2 生成与安装

生成格式文件和文档:

1. 对 T_EX 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`: 字体名映射文件。

对 MiKTeX 等发行版, 执行

```
texlua zhmCJK.lua nomap
```

生成目录 `fallback` 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 `fallback/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。编译文档需要系统安装有 `simsun.ttc` (宋体) 和 `simkai.ttf` (楷体) 两种中文 Windows 预装字体。

设 `TEXMF` 是 $\text{T}_{\text{E}}\text{X}$ 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.txt
TEXMF/fonts/map/fontname/texfonts.map
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
```

如果目录中原来已有非空的 `texfonts.map`，可以把 `zhmCJK` 所附的 `texfonts.map` 文件内容附加在原有的 `texfonts.map` 之后。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 `fallback` 目录下的所有文件：

```
TEXMF/fonts/tfm/zhmCJK/fallback/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 fails,
41     try 'elatex' or 'pdfelatex' command.
42     If all fails, 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 use lualatex or dvilualatex to compile
48     the file. zhmCJK do 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

```
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。

```
120 \AtEndOfPackage{\CJK@makeActive}
121 \edef\document{%
122   \unexpanded\expandafter{\document}%
123   \unexpanded{\begin{CJK*}{\zhm@encoding}{\CJKfamilydefault}}}
124 \edef\enddocument{%
125   \unexpanded{\clearpage\end{CJK*}}%
126   \unexpanded\expandafter{\enddocument}}
```

`\zhm@fontnum` 已定义的 CJK 字体数。

```
127 \newcount\zhm@fontnum
```

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

```
128 \ifpdf
129   \def\zhm@pdfliteral#1{\pdfliteral{#1}}
130 \else
131   \def\zhm@pdfliteral#1{\special{pdf:literal #1}}
132 \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。

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

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

`\zhm@pdf@endbold`

```
140 \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。

```
141 \def\zhm@zero{0}
142 \def\ifzhm@fakebold{\expandafter\unless\expandafter\ifx
143   \csname zhm@\zhm@fam @embolden\endcsname
144   \zhm@zero}
145 \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：

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

`\zhm@pdf@beginslant`

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

`\zhm@pdf@endslant`

```
148 \def\zhm@pdf@endslant{%
149   \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。

```
150 \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。

```
151 \newif\ifzhm@CJK@slant@
```

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

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

`\CJKslant` 设置伪斜体（不加粗）。

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

`\CJKboldslant` 设置伪粗体加伪斜体。

```

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

```

`\CJKnormal` 恢复无伪粗、伪斜体的状态。

```

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

```

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

```

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

```

188 \def\zhm@DeclareFontShape#1#2#3#4#5#6{%
189   \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。

```

190 \DeclareSizeFunction{CJKsl}{\CJK@empty\@font@warning}
191 \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]。

```

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

```

199 \def\zhm@setfontmap#1#2{%
200   \AtBeginDvi{\zhm@mapline{#1}{#2}}
201   \AtBeginDocument{%
202     \ifpackageloaded{atbegshi}{\AtBeginShipoutFirst{%
203       \zhm@mapline{#1}{#2}}}{}}
```

`\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。

```

204 \def\zhm@DeclareRealFont#1{
205   \unless\ifcsname zhm@#1@fontnum\endcsname
206     \global\advance\zhm@fontnum\@ne
207     \ifnum\zhm@fontnum>\@xxxii
208       \PackageError{zhmCJK}%
209         {No more CJK font families can be setup.}%
210         {There are at most 32 families setup by zhmCJK.}
211     \else
212       \expandafter\xdef\csname zhm@#1@fontnum\endcsname{\the\zhm@fontnum}
213       \zhm@setfontmap{\the\zhm@fontnum}{#1}
214     \fi
215 \fi
```



```

216 \edef\zhm@tfm{zhm\@nameuse{zhm#1@fontnum}}
217 \define@key{zhm}{cmap}{%
218 \edef\zhm@cmap{#1}}
219 \define@key{zhm}{embed}[true]{%
220 \ifpdf
221 \PackageWarning{zhmCJK}
222 {The ‘embed’ option is not supported by pdfTeX driver.\MessageBreak
223 Ignoring option ‘embed’}
224 \fi
225 \edef\zhm@embed{#1}%
226 \ifx\zhm@embed\zhm@true
227 \zhm@embedtrue
228 \else\ifx\zhm@embed\zhm@false
229 \zhm@embedfalse
230 \else
231 \PackageError{zhmCJK}%
232 {The ‘embed’ option must be true or false.}%
233 {See the package manual for more information.}%
234 \fi\fi}
235 \def\zhm@true{true}
236 \def\zhm@false{false}
237 \define@key{zhm}{encoding}{%
238 \edef\zhm@encoding{#1}%
239 \zhm@check@encoding
240 \edef\zhm@nfssenc{\@nameuse{CJK@\zhm@encoding @nfssenc}}%
241 \edef\zhm@sfd{\@nameuse{zhm@\zhm@encoding @sfd}}%
242 \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。

```

243 \define@key{zhm}{AutoFakeBold}{%
244 [\ifx\zhm@AutoFakeBold\zhm@zero 0.4\else\zhm@AutoFakeBold \fi]%
245 {\expandafter\xdef\csname zhm@\zhm@fam @embolden\endcsname{#1}}

```

`\zhm@neg` 利用长度计算取负数。

```

246 \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。

```

247 \define@key{zhm}{AutoFakeSlant}{%
248 [\ifx\zhm@AutoFakeSlant\zhm@zero 0.167\else\zhm@AutoFakeSlant \fi]%

```

```

249 {\expandafter\xdef\csname zhm@\zhm@fam @slant\endcsname{#1}%
250 \expandafter\xdef\csname zhm@\zhm@fam @negslant\endcsname{\zhm@neg{#1}}}

```

设置 BoldFont 选项。

```

251 \define@key{zhm}{BoldFont}{
252   \zhm@DeclareRealFont{#1}
253   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{n}{<-> CJK * \zhm@tfm}%
254   {\CJKnormal}
255   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{n}{<-> CJK * \zhm@tfm}%
256   {\CJKnormal}
257   \zhm@fakeboldfalse}

```

设置 ItalicFont 选项。

```

258 \define@key{zhm}{ItalicFont}{
259   \zhm@DeclareRealFont{#1}
260   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{it}{<-> CJK * \zhm@tfm}%
261   {\CJKnormal}}

```

设置 BoldItalicFont 选项。

```

262 \define@key{zhm}{BoldItalicFont}{
263   \zhm@DeclareRealFont{#1}
264   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{it}{<-> CJK * \zhm@tfm}%
265   {\CJKnormal}
266   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{it}{<-> CJK * \zhm@tfm}%
267   {\CJKnormal}}

```

设置 SlantedFont 选项。

```

268 \define@key{zhm}{SlantedFont}{
269   \zhm@DeclareRealFont{#1}
270   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{sl}{<-> CJK * \zhm@tfm}%
271   {\CJKnormal}
272   \zhm@fakeslantfalse}

```

设置 BoldSlantedFont 选项。

```

273 \define@key{zhm}{BoldSlantedFont}{
274   \zhm@DeclareRealFont{#1}
275   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{sl}{<-> CJK * \zhm@tfm}%
276   {\CJKnormal}
277   \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{sl}{<-> CJK * \zhm@tfm}%
278   {\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。

```

279 \def\zhm@newfontfamily#1[#2]#3{

```

280 \beginngroup

警告字体族重定义

281 \ifcsname zhm@#1@font\endcsname

282 \PackageWarning{zhmCJK}{Redefinition of CJK family ‘#1’}

283 \fi

缓存字体族名及对应主字体文件名。

284 \edef\zhm@fam{#1}

285 \expandafter\xdef\csname zhm@#1@font\endcsname{#3}

使全局的 AutoFakeBold 和 AutoFakeSlant 生效。

286 \setkeys{zhm}{%

287 AutoFakeBold=\zhm@AutoFakeBold,

288 AutoFakeSlant=\zhm@AutoFakeSlant}

定义 CJK 字体族。

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

执行选项，定义编码、CMap 和其他字形等。

290 \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]。

291 \zhm@DeclareRealFont{#3}

292 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{n}{<-> CJK * \zhm@tfm}{%

293 {\CJKnormal}}

按需要，定义伪粗体、伪斜体字形。

294 \ifzhm@fakebold

295 \zhm@DeclareRealFont{#3}

296 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{n}{%

297 {<-> CJKb * \zhm@tfm}{\CJKbold}}

298 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{n}{%

299 {<-> CJKb * \zhm@tfm}{\CJKbold}}

300 \fi

301 \ifzhm@fakeslant

302 \zhm@DeclareRealFont{#3}

303 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{m}{sl}{%

304 {<-> CJKsl * \zhm@tfm}{\CJKslant}}

305 \fi

306 \expandafter\ifzhm@fakeslant\ifzhm@fakebold

307 \zhm@DeclareRealFont{#3}

308 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{bx}{sl}{%

309 {<-> CJKbsl * \zhm@tfm}{\CJKboldslant}}

310 \zhm@DeclareFontShape{\zhm@nfssenc}{\zhm@fam}{b}{sl}{%

```

311      {<-> CJKbsl * \zhm@tfm}{\CJKboldslant}
312  \fi\fi
313  \endgroup}

```

`\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。

```

314 \newcommand*\setCJKfamilyfont[1]{%
315   \@ifnextchar[%
316     {\zhm@newfontfamily{#1}}%
317     {\zhm@newfontfamily{#1}[]}}
318 \@onlypreamble\setCJKfamilyfont

```

`\setCJKmainfont` 设置 CJK 普通（罗马）字体。

```

319 \newcommand*\setCJKmainfont[2] []{%
320   \zhm@newfontfamily{\CJKrmdefault}[#1]{#2}}
321 \@onlypreamble\setCJKmainfont

```

`\setCJKromanfont` `\setCJKmainfont` 的别名。

```

322 \let\setCJKromanfont\setCJKmainfont
323 \@onlypreamble\setCJKromanfont

```

`\setCJKsansfont` 设置 CJK 无衬线字体。

```

324 \newcommand*\setCJKsansfont[2] []{%
325   \zhm@newfontfamily{\CJKsfdefault}[#1]{#2}}
326 \@onlypreamble\setCJKsansfont

```

`\setCJKmonofont` 设置 CJK 等宽（打字机）字体。

```

327 \newcommand*\setCJKmonofont[2] []{%
328   \zhm@newfontfamily{\CJKttdefault}[#1]{#2}}
329 \@onlypreamble\setCJKmonofont

```

`\CJKrmdefault` CJK 罗马体默认字体族，作用于 `\rmfamily`。

```

330 \providecommand*\CJKrmdefault{rm}

```

`\CJKsfdefault` CJK 无衬线体默认字体族，作用于 `\sffamily`。

```

331 \providecommand*\CJKsfdefault{sf}

```

`\CJKttdefault` CJK 打字机体默认字体族，作用于 `\ttfamily`。

```

332 \providecommand*\CJKttdefault{tt}

```

`\CJKfamilydefault` CJK 默认字体族，作用于 `\normalfont`。

```

333 \providecommand*\CJKfamilydefault{\CJKrmdefault}

```

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

```
334 \DeclareRobustCommand\normalfont
335     {\CJKfamily{\CJKfamilydefault}}%
336     \usefont\encodingdefault
337         \familydefault
338         \seriesdefault
339         \shapedefault
340     \relax}
341 \let\reset@font\normalfont
342 \DeclareRobustCommand\rmfamily
343     {\not@math@alphabet\rmfamily\mathrm
344     \fontfamily\rmdefault\CJKfamily{\CJKrmdefault}\selectfont}
345 \DeclareRobustCommand\sffamily
346     {\not@math@alphabet\sffamily\mathsf
347     \fontfamily\sfdefault\CJKfamily{\CJKsfdefault}\selectfont}
348 \DeclareRobustCommand\ttfamily
349     {\not@math@alphabet\ttfamily\mathtt
350     \fontfamily\ttdefault\CJKfamily{\CJKttdefault}\selectfont}
351 \end{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 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.
```

¹中科院应用数学研究所副研究员。wulingyun@gmail.com

²zwhuang@gmail.com

```

14  --
15  -- This work consists of the files zhmCJK.dtx,
16  --                                zhmCJK.ins,
17  --                                zhmCJK.lua,
18  --          and the derived files zhmCJK.sty,
19  --                                zhmCJK.pdf,
20  --                                zhmCJK-test.tex,
21  --                                README.txt,          (from zhmCJK.dtx)
22  --                                zhmCJK.map,
23  --                                texfonts.map,
24  --                                zhmCJK.tfm,
25  --                                fallback/zhm*/zhm**.tfm. (from zhmCJK.lua)
26  --
27  -----
28  -- This lua script is used to generate zhm*.tfm and zhmCJK.map.
29  --
30  -- Usage:
31  --
32  --     texlua zhmCJK.lua map
33  --
34  -- or (for MiKTeX only)
35  --
36  --     texlua zhmCJK.lua nomap
37  --
38  --
39  -----
40  -- OS dependent settings
41  -----
42  --
43  if os.type == "unix" then
44     path_slash = "/"
45  else
46     path_slash = "\\"
47  end
48  --
49  -----
50  -- Generate TFM
51  -----
52  --
53  pl_template = {
54     char = [[

```

```

55 (CHARACTER D %d
56     (CHARWD R 1.0)
57     (CHARHT R 0.8)
58     (CHARDP R 0.1)
59 )
60 ]],
61     header = [[
62 (FAMILY %s)
63 (CODINGScheme CJK-UNICODE)
64 (DESIGNSIZE R 10.0)
65 (HEADER D 18 H 43726561)
66 (HEADER D 19 H 74656420)
67 (HEADER D 20 H 62792060)
68 (HEADER D 21 H 7A686D43)
69 (HEADER D 22 H 4A4B2E6C)
70 (HEADER D 23 H 7561272E)
71 (HEADER D 24 H 20436F70)
72 (HEADER D 25 H 79726967)
73 (HEADER D 26 H 68742028)
74 (HEADER D 27 H 43292032)
75 (HEADER D 28 H 30313220)
76 (HEADER D 29 H 6279204C)
77 (HEADER D 30 H 656F204C)
78 (HEADER D 31 H 6975203C)
79 (HEADER D 32 H 6C656F6C)
80 (HEADER D 33 H 69752E70)
81 (HEADER D 34 H 6B754067)
82 (HEADER D 35 H 6D61696C)
83 (HEADER D 36 H 2E636F6D)
84 (HEADER D 37 H 3E0D0A00)
85 (CHECKSUM 0 0)
86 (FONTDIMEN
87     (SLANT R %f)
88     (SPACE R 0.5)
89     (STRETCH R 0.3)
90     (SHRINK R 0.1)
91     (XHEIGHT R 0.4)
92     (QUAD R 1.0)
93 )
94 ]],
95 -- charset = (defined below)

```

```

96 }
97
98 function pl_charset()
99     local charset = {}
100     for cid = 0, 0xff do
101         charset[#charset + 1] = string.format(pl_template.char, cid)
102     end
103     return table.concat(charset)
104 end
105
106 pl_template.charset = pl_charset()
107
108 function write_tfm(path, name, slant)
109     local fullname = path .. path_slash .. name
110     local s_pl = string.format(pl_template.header, string.upper(name), slant)
111         .. pl_template.charset
112     local f_pl = io.open(fullname .. ".pl", "w")
113     f_pl:write(s_pl)
114     f_pl:close()
115     os.execute([[pltotf ]] .. fullname .. [[.pl" ]] .. fullname .. [[.tfm]])
116     os.remove(fullname .. ".pl")
117 end
118
119 -----
120 -- Main functions to generate necessary files
121 -----
122
123 -- For TeX Live and other TeX distributions where texfonts.map is supported,
124 -- we generate texfonts.map, zhmCJK.map, and zhmCJK.tfm.
125 function generate_with_fontmap()
126     local f_map = io.open("zhmCJK.map", "w")
127     for fam = 1, 32 do
128         for sid = 0, 0xff do
129             f_map:write(string.format("zhmCJK.tfm zhm%d%02x.tfm\n",
130                 fam, sid))
131         end
132     end
133     f_map:write("\n")
134     f_map:close()
135
136     f_map = io.open("texfonts.map", "w")

```



```

137     f_map:write("include zhmCJK.map\n\n")
138     f_map:close()
139
140     write_tfm(".", "zhmCJK", 0.0)
141 end
142
143 -- For MiKTeX and other TeX distributions where texfonts.map is not supported,
144 -- we generate a lot of zhmXYY.tfm, where X from 1 to 32, Y from 0x00 to 0xff.
145 function generate_without_fontmap()
146     lfs.mkdir("fallback")
147     for fam = 1, 32 do
148         local path = string.format("fallback" .. path_slash .. "zhm%d", fam)
149         lfs.mkdir(path)
150         print(path)
151         for sid = 0, 0xff do
152             local name = string.format("zhm%d%02x", fam, sid)
153             write_tfm(path, name, 0.0)
154         end
155     end
156 end
157
158 -----
159 -- Console User Interface
160 -----
161
162 help_info = [[
163 Usage:
164
165     texlua ]].. arg[0] .. [[ map|nomap
166
167     map:    Generate a public TFM shared by all CJK fonts with mapping files.
168             It is suggested for TeX Live and other web2c distributions.
169
170     nomap:  Generate all TFM files for CJK fonts into "fallback" directory.
171             MiKTeX needs this since it does not support TFM mappings.
172 ]]
173
174 if #arg ~= 1 then
175     print(help_info)
176 else
177     if arg[1] == "map" then

```

```

178         generate_with_fontmap()
179     elseif arg[1] == "nomap" then
180         generate_without_fontmap()
181     else
182         print("! Unknown option " .. arg[1])
183         print(help_info)
184     end
185 end
186
187 -- end of file zhmCJK.lua --
188 --
189 -----
190 --
191 -- The code is inspired by zhtfm.lua of zhmetrics bundle.
192 --
193 -- The zhmetrics bundle is distributed under LPPL 1.3 or later. The original
194 -- authors of zhtfm.lua are:
195 --
196 --         Lingyun Wu     <wulingyun@gmail.com>
197 --         zwhuang       <zwhuang@gmail.com>
198 --
199 -- For more information of zhmetrics, see the ctex-kit project:
200 --
201 --         http://code.google.com/p/ctex-kit/
202 --
203

```

6 版本历史

v0.1		\setCJKromanfont: 新增. . .	20
General: 初始版本	1	v0.4	
v0.2		General: 增加宏包 AutoFakeBold	
General: 编写宏包文档。增加		选项	10
CJKpunct。做一些小的代码		设置字体 AutoFakeBold 选	
调整。	1	项。	17
v0.3		设置字体 BoldFont 选项。 .	18
General: 使用 PDF 原语生成伪		设置字体 BoldItalicFont 选	
粗体	13	项。	18
增加宏包 pdffakebold 选项	10	设置字体 BoldSlantedFont	

选项。	18	参数位置, 以与 xeCJK 包语	
设置字体 <code>ItalicFont</code> 选项。	18	法一致。	20
设置字体 <code>SlantedFont</code> 选项。		<code>\zhm@newfontfamily</code> : 更改语	
.	18	法, 修改选项位置。	18
v0.5		v0.8	
General: 使用 Lua 脚本生成		General: 增加宏包 <code>encoding</code> 选	
TFM 和映射文件; 为 MiK-		项选择默认编码。	10
TeX 生成单独的 TFM 文件。		设置字体 <code>encoding</code> 选项。 .	17
将原来的 <code>zhmetrics.tfm</code> 改		v0.9	
名为 <code>zhmCJK.tfm</code> , 不再依赖		General: 增加宏包 <code>cmap</code> 选项选	
原有的 <code>zhmetrics</code> 包。	1	择 DVIPDFMx 的 CMap 映	
v0.6		射。	10
General: 增加字体 <code>AutoFakeSlant</code>		增加宏包 <code>embed</code> 选项, 允许	
选项	17	DVIPDFMx 驱动不嵌入字	
增加宏包 <code>AutoFakeSlant</code> 选		体。	10
项	10	支持伪粗斜体。	16
v0.7		设置字体 <code>cmap</code> 选项。	17
General: 宏包载入时进行编译引		设置字体 <code>embed</code> 选项。	17
擎测试	9	重新实现伪粗体、伪斜体功	
生成 TFM 时加入版权说明。 26		能, 使之可以正常处理断行、	
<code>\setCJKfamilyfont</code> : 修改可选		标点压缩等问题。	15

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; 而直立体的数字表示对应项使用时所在的行号。

A	<code>\CJK@family</code>	<code>\CJKnormal</code> 160, 165,
<code>\AddToKeyvalOption</code>	. . . 139, 147, 149	<u>167</u> , 254, 256,
.	67	<u>261</u> , 265, 267,
<code>AutoFakeBold</code> (op-	<code>\CJK@global</code> 93	271, 276, 278, 293
tion)	<code>\CJK@local</code> 95	<code>\CJKrmdefault</code> . . 4,
<code>AutoFakeSlant</code> (op-	<code>\CJK@makeActive</code> . 120	320, <u>330</u> , 333, 344
tion)	<code>\CJKbold</code> <u>152</u> , 297, 299	<code>\CJKsfdefault</code>
	<code>\CJKboldshift</code> 183, 184	. 4, 325, <u>331</u> , 347
C	<code>\CJKboldslant</code>	<code>\CJKslant</code> . . . <u>157</u> , 304
<code>\CJK@active</code> 94	. . . <u>162</u> , 309, 311	<code>\CJKsymbol</code> <u>178</u>
<code>\CJK@bold@false</code> . .	<code>\CJKfamily</code> . . . 26,	<code>\CJKttdefault</code>
. . . 160, 165, 170	335, 344, 347, 350	<code>\CJKttdefault</code>
<code>\CJK@bold@true</code> . . 155	<code>\CJKfamilydefault</code>	. 4, 328, <u>332</u> , 350
<code>\CJK@empty</code> . . 190, 191	. 4, 123, <u>333</u> , 335	<code>cmap</code> (option) <u>65</u>

D	<code>\pdfliteral</code> . 129, 134	<code>\zhm@CJK@bold@false</code> 158, 168
<code>\DeclareFontFamily</code> 289	<code>\pdfmapline</code> 194	<code>\zhm@CJK@bold@true</code> 153, 163
<code>\DeclareFontShape</code> 189	R	<code>\zhm@CJK@slant@false</code> 153, 155, 160, 165, 168, 170
<code>\DeclareSizeFunction</code> 190, 191	<code>\reset@font</code> 341	<code>\zhm@CJK@slant@true</code> 158, 163
<code>\document</code> ... 121, 122	<code>\rmdefault</code> 344	<code>\zhm@cmmap</code> ... 197, 218
	<code>\rmfamily</code> ... 342, 343	<code>\zhm@DeclareFontShape</code> 188, 253, 255, 260, 264, 266, 270, 275, 277, 292, 296, 298, 303, 308, 310
E	S	<code>\zhm@DeclareRealFont</code> 204, 252, 259, 263, 269, 274, 291, 295, 302, 307
<code>embed (option)</code> 66	<code>\selectfont</code> 344, 347, 350	<code>\zhm@def@sfd</code> 102, 103, 104, 105, 106, 107, 108, 109
<code>encoding (option)</code> . 73	<code>\seriesdefault</code> .. 338	<code>\zhm@embed</code> 225, 226, 228
<code>\encodingdefault</code> 336	<code>\setCJKfamilyfont</code> 3, 10, 314	<code>\zhm@embedfalse</code> . 229
<code>\enddocument</code> 124, 126	<code>\setCJKmainfont</code> 2, 7, 319, 322	<code>\zhm@embedtrue</code> .. 227
F	<code>\setCJKmonofont</code> 3, 9, 327	<code>\zhm@encoding</code> . 80, 82, 86, 91, 100, 101, 111, 113, 123, 238, 240, 241
<code>\familydefault</code> .. 337	<code>\setCJKromanfont</code> 3, 322	<code>\zhm@fakeboldfalse</code> 145, 257
<code>\fontfamily</code> 344, 347, 350	<code>\setCJKsansfont</code> 3, 8, 324	<code>\zhm@fakeboldtrue</code> 175, 272
I	<code>\sfdefault</code> 347	<code>\zhm@false</code> .. 228, 236
<code>\ifCJK@bold@</code> ... 182	<code>\sffamily</code> ... 345, 346	<code>\zhm@fam</code> . 143, 145, 173, 176, 177, 245, 249, 250, 253, 255, 260, 264, 266, 270,
<code>\ifzhm@CJK@bold@</code> 150, 180, 186	<code>\shapedefault</code> ... 339	
<code>\ifzhm@CJK@slant@</code> ... 151, 179, 187	<code>\special</code> 131, 136, 197	
<code>\ifzhm@embed</code> ... 197	T	
<code>\ifzhm@fakebold</code> 141, 294, 306	<code>\ttdefault</code> 350	
<code>\ifzhm@fakeslant</code> 172, 301, 306	<code>\ttfamily</code> ... 348, 349	
<code>\ifzhm@pdffakebold</code> 152, 157, 162, 167	U	
<code>\ifzhm@primitive</code> 31, 36, 45, 52	<code>\usefont</code> 336	
M	Z	
<code>\MessageBreak</code> 70, 222	<code>\zhm@AutoFakeBold</code> 244, 287	
N	<code>\zhm@AutoFakeSlant</code> 248, 288	
<code>\normalfont</code> . 334, 341	<code>\zhm@check@encoding</code> 79, 88, 239	
P	<code>\zhm@check@sfd</code> 110, 242	
<code>pdffakebold (option)</code> 74		

	275, 277, 284,	270, 275, 277,	... 133, 139, 140
	289, 292, 296,	289, 292, 296,	\zhm@setfontmap ..
	298, 303, 308, 310	298, 303, 308, 310 199, 213
\zhm@fontnum	127,	\zhm@pdf@beginbold	\zhm@sfd 101,
	206, 207, 212, 213 138, 180	117, 194, 197, 241
\zhm@mapline	\zhm@pdf@beginslant	\zhm@tfm . 216, 253,
	... 192, 200, 203 146, 179	255, 260, 264,
\zhm@neg 246, 250	\zhm@pdf@endbold .	266, 270, 275,
\zhm@newfontfamily	 140, 186	277, 292, 297,
 279, 316,	\zhm@pdf@endslant	299, 304, 309, 311
	317, 320, 325, 328 148, 187	\zhm@true ... 226, 235
\zhm@nfssenc	100,	\zhm@pdfliteral ..	\zhm@UTF 90, 91
	240, 253, 255,	... 128, 147, 149	\zhm@zero ... 141,
	260, 264, 266,	\zhm@pdfliteraldirect	144, 174, 244, 248

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>
- [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>
- [5] WERNER LEMBERG. *The fonts*, 2008
<CTAN://language/chinese/CJK/cjk-4.8.2/doc/fonts.txt>
- [6] HÀN THẾ THÀNH. “A closer look at TrueType fonts and pdf_TE_X”. *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>

- [8] 王越. *CJK* 宏包中, 中文字体的秘密. CT_EX 论坛, 2009 年 5 月
<http://bbs.ctex.org/viewthread.php?tid=50078>