

# xeLaTeX-xeCJK 中文字体使用方法简要

tsingke

2020 年 2 月 27 日

## 1 前言

XETEX 可以通过 `\fontspec{}` 及 `\selectfont` 指令调用系统本地字体（包括中文字体），但如果使用 CTeX 宏包，这条指令就对中文字符无效；而如果不使用 CTeX 宏包，输入的中文又无法断句。针对这种两难局面，可以采用 X<sub>EL</sub>TEX + CTeX 中使用系统本地字体的方法，使用户可以设定中文字符的字体。

## 2 什么是 XET<sub>E</sub>X

XETEX 是一种使用 Unicode 的 TeX 排版引擎，并支持一些现代字体技术。XETEX 最初只是为 Mac OS X 所开发，但它如今在各主要平台上都可以运作。它原生支持 Unicode，并默认其输入文件为 UTF-8 编码。XETEX 可以在不进行额外配置的情况下直接使用操作系统中安装的字体，因此可以直接利用 OpenType、Graphite 中的高级特性，例如额外的字形，花体，合字，可变的文本粗细等等。XETEX 提供了对 OpenType 中本地排版约定（locl 标签）的支持，也允许向字体传递 OpenType 的元标签。它亦支持使用包含特殊数学字符的 Unicode 字体排版数学公式，例如使用 Cambria Math 或 Asana Math 字体代替传统的 TeX 字体。目前使用较为广泛的是 XET<sub>E</sub>X，它底层继承自 XET<sub>E</sub>X，提供了更方便易用的特性。

## 3 xeCJK 解析

xeCJK 是一个 X<sub>EL</sub>AT<sub>E</sub>X 的宏包，专门用于排版中日韩文字（CJK 含义：Chinese, Japanese, Korean，简称 CJK）。xeCJK 的原始作者是孙文昌，2009 年 5 月起宏包被收入 ctex-ki 项目进行维护，目前主要维护者是刘海洋和李清。xeCJK 只提供了字体和标点控制等基本 CJK 语言支持。对于中文文档，可以使用更为高层的 ctex 宏包或文档类，它将自动调用 xeCJK 并设置好中文字体，同时提供了进一步的本地化支持。详细内容参看 ctex 宏包套件的说明。与其他 L<sup>A</sup>T<sub>E</sub>X 宏包一样，引入 xeCJK 宏包只要要在导言区使用，

xeCJK 提供了大量选项,可以在宏包调用时作为宏包选项或用 `xeCJKsetup` 命令进行设置。除了 `setCJKmainfont` 命令,xeCJK 还提供了许多其他命令设置和选择中文字体。此外,在使用 X<sub>Y</sub>TEX 时 listings 宏包可以支持 Unicode,因此在 listings 定义的代码环境中可以直接使用中文。

xeCJK 对标点符号的输出宽度的调整是通过调整其左边或右边的空白宽度来实现的。按照目前的处理方式,对于位于左边的标点符号(如左引号),xeCJK 只能调整它左边的空白;对于位于右边的标点符号(如右引号),xeCJK 只能调整它右边的空白;对于居中的标点符号,则调整其左右空白,以保证其居中。对于标点符号的相关设置,只能在导言区中进行。X<sub>Y</sub>TEX 宏包的主要功能:

1. 分别设置 CJK 和英文字体;
2. 自动忽略 CJK 文字间的空格而保留其它空格,允许在非标点汉字和英文字母 (a -z, A -Z) 间断行;
3. 提供多种标点处理方式:全角式、半角式、开明式、行末半角式和 CCT 式;
4. 自动调整中英文间空白。

xeCJK 使用了 X<sub>Y</sub>TEX 的一些最新特性,xeCJK 依赖 L<sup>A</sup>T<sub>E</sub>X3 项目的宏包套件 l3kernel 和 l3packages。xeCJK 还需要通过 fontspec 宏包来调用系统字体。xeCJK 会自动根据需要载入这些宏包。

### 3.1 如何通过 xecjk 调用中文字体

所谓设置全局字体就是在 X<sub>Y</sub>TEX 源码的导言区使用命令对该作品中所有文字的字体做统一设定,命令如下:`\setmainfont{英文字体名称}`  
`\setCJKmainfont{中文字体名称}`

上面两个命令的区别是 `\setmainfont` 设定的是英文字体,`\setCJKmainfont` 设定的是 CJK 字符字体(CJK 字符包括中文、日文、韩文等,CJK 就是中日韩三国英文国名的首字母缩写)。两个命令可以同时存在、互不影响,也就是说可以分别设定英文和中文的字体。理论上讲凡是使用 `fc-list` 命令查到的字体名称都可以作为这两个命令的参数,即都用这两个命令设定为全局字体,但实际中可能会有一些字体因为其名称的问题而不能采用这种方法(比如汉仪的一些字体),所幸这样的奇葩字体少之又少。

如果深入探究的话,你会发现类似于以上两个命令的还有 `\setsansfont`、`\setmonofont`、`\setCJKsansfont` 和 `\setCJKmonofont`,它们的含义分别是设置全局西文无衬线字体、设置全局西文等宽字体、设置全局 CJK 无衬线字体和设置全局 CJK 等宽字体。对于需要看这篇文章的同学们来说,可以简单认为这四

个命令没什么用，因为它们涉及到更加专业的排版技术细节，有些甚至涉及打印机的 DPI 参数。如果你一定要了解清楚的话，建议自行 Google;

大家知道在 $\text{\LaTeX 2}_{\epsilon}$ 中可以使用  $\{\backslash \text{字体名称正文内容}\}$  这样的格式来实现类似的效果，但这一指令本质上是使用 $\text{\LaTeX 2}_{\epsilon}$ 中内置的字体，不是本地系统的字体，因此可选的也只有 heiti、SimSun 等五六种少得可怜的中文字体，tex 系统内置的**六大字体**：宋体、仿宋、楷书、~~隶书~~、黑体、幼圆

X $\text{\LaTeX}$  在这方面要比 $\text{\LaTeX 2}_{\epsilon}$ 强大得多，毕竟它使用的是 Unicode 编码；但是在 X $\text{\LaTeX}$  中直接使用  $\{\backslash \text{字体名称正文内容}\}$  这样的指令，想简单地把字体名称替换成本地字体名称是不行。必须首先在导言区使用以下指令设定相关命令：

设定中文字体指令：

$\backslash\text{setCJKfamilyfont}\{\text{自定义的 CJKfamily 名称}\}\text{系统字体名称}$

$\backslash\text{newcommand}\{\text{自定义指令名}\}\{\text{自定义的 CJKfamily 名称}\}$

举例说明，如何使用上述两条命令实现对本地系统内字体的使用：

$\backslash\text{setCJKfamilyfont}\{\text{hwsk}\}\{\text{STXingkai}\}$

$\backslash\text{newcommand}\{\backslash\text{stxk}\}\{\backslash\text{CJKfamily}\{\text{hwsk}\}\}$

按照上面的步骤设定好之后，就可以像 L $\text{\TeX}$  那样使用 字体名称正文内容 命令来设定字体，你可以用这样的大括号对把整个一个章节都括起来，也可以只括住某个词或字母，如此就可以非常灵活方便地控制字体，可以实现在全文使用某一字体的前提下，在某一特定段落使用另一字体。

### 3.2 xeCJK 具体设置方法命令

```
%—————添加 windows 本地系统内的字体—————  
 $\backslash\text{setCJKfamilyfont}\{\text{song}\}\{\text{SimSun}\}$  %宋体 song  
 $\backslash\text{newcommand}\{\backslash\text{song}\}\{\backslash\text{CJKfamily}\{\text{song}\}\}$  % 宋体 (Windows 自带 sim sun. ttf)  
  
 $\backslash\text{setCJKfamilyfont}\{\text{xs}\}\{\text{NSimSun}\}$  %新宋体 xs  
 $\backslash\text{newcommand}\{\backslash\text{xs}\}\{\backslash\text{CJKfamily}\{\text{xs}\}\}$   
  
 $\backslash\text{setCJKfamilyfont}\{\text{fs}\}\{\text{FangSong\_GB2312}\}$  %仿宋 2312 fs  
 $\backslash\text{newcommand}\{\backslash\text{fs}\}\{\backslash\text{CJKfamily}\{\text{fs}\}\}$  %仿宋体 (Windows 自带 sim fs. ttf)  
  
 $\backslash\text{setCJKfamilyfont}\{\text{kai}\}\{\text{KaiTi\_GB2312}\}$  %楷体 2312 kai  
 $\backslash\text{newcommand}\{\backslash\text{kai}\}\{\backslash\text{CJKfamily}\{\text{kai}\}\}$   
  
 $\backslash\text{setCJKfamilyfont}\{\text{yh}\}\{\text{Microsoft YaHei}\}$  %微软雅黑 yh  
 $\backslash\text{newcommand}\{\backslash\text{yh}\}\{\backslash\text{CJKfamily}\{\text{yh}\}\}$ 
```

```

\setCJKfamilyfont{hei}{SimHei} %黑体 hei
\newcommand{\hei}{\CJKfamily{hei}} %黑体 (Windows自带 simhei.ttf)

\setCJKfamilyfont{msunicode}{Arial Unicode MS} %Arial Unicode MS: msun
\newcommand{\msunicode}{\CJKfamily{msunicode}}

\setCJKfamilyfont{li}{LiSu} %隶书 li
\newcommand{\li}{\CJKfamily{li}}

\setCJKfamilyfont{yy}{YouYuan} %幼圆 yy
\newcommand{\yy}{\CJKfamily{yy}}

\setCJKfamilyfont{xm}{MingLiU} %细明体 xm
\newcommand{\xm}{\CJKfamily{xm}}

\setCJKfamilyfont{xxm}{PMingLiU} %新细明体 xxm
\newcommand{\xxm}{\CJKfamily{xxm}}

\setCJKfamilyfont{hwsong}{STSong} %华文宋体 hwsong
\newcommand{\hwsong}{\CJKfamily{hwsong}}

\setCJKfamilyfont{hwzs}{STZhongsong} %华文中宋 hwzs
\newcommand{\hwzs}{\CJKfamily{hwzs}}
\setCJKfamilyfont{hwfs}{STFangsong} %华文仿宋 hwfs
\newcommand{\hwfs}{\CJKfamily{hwfs}}

\setCJKfamilyfont{hwxxh}{STXihei} %华文细黑 hwxxh
\newcommand{\hwxxh}{\CJKfamily{hwxxh}}

\setCJKfamilyfont{hwsl}{STLiti} %华文隶书 hwsl
\newcommand{\hwsl}{\CJKfamily{hwsl}}

\setCJKfamilyfont{hwxxw}{STXinwei} %华文新魏 hwxxw
\newcommand{\hwxxw}{\CJKfamily{hwxxw}}

\setCJKfamilyfont{hwk}{STKaiti} %华文楷体 hwk
\newcommand{\hwk}{\CJKfamily{hwk}}

```

```
\setCJKfamilyfont{hwk}{STXingkai} %华文行楷 hwk
\newcommand{\hwk}{\CJKfamily{hwk}}
```

```
\setCJKfamilyfont{hwcy}{STCaiyun} %华文彩云 hwcy
\newcommand{\hwcy}{\CJKfamily{hwcy}}
```

```
\setCJKfamilyfont{hwhp}{STHupo} %华文琥珀 hwhp
\newcommand{\hwhp}{\CJKfamily{hwhp}}
```

```
\setCJKfamilyfont{fzsong}{Simsun (Founder Extended)} %方正宋体超大字符集
\newcommand{\fzsong}{\CJKfamily{fzsong}}
```

```
\setCJKfamilyfont{fzyao}{FZYaoTi} %方正姚体 fzy
\newcommand{\fzyao}{\CJKfamily{fzyao}}
```

```
\setCJKfamilyfont{fzshu}{FZShuTi} %方正舒体 fzshu
\newcommand{\fzshu}{\CJKfamily{fzshu}}
```

```
\setCJKfamilyfont{asong}{Adobe Song Std} %Adobe 宋体 asong
\newcommand{\asong}{\CJKfamily{asong}}
```

```
\setCJKfamilyfont{ahei}{Adobe Heiti Std} %Adobe 黑体 ahei
\newcommand{\ahei}{\CJKfamily{ahei}}
```

```
\setCJKfamilyfont{akai}{Adobe Kaiti Std} %Adobe 楷体 akai
\newcommand{\akai}{\CJKfamily{akai}}
```

%—————标题名称中文化—————%

```
\renewcommand\abstractname{\hei 摘\要}
```

```
\renewcommand\refname{\hei 参考文献}
```

```
\renewcommand\figurename{\hei 图}
```

```
\renewcommand\tablename{\hei 表}
```

%—————定理名称中文化—————%

```
\newtheorem{dingyi}{\hei 定义~}[section]
```

```
\newtheorem{dingli}{\hei 定理~}[section]
```

```
\newtheorem{yinli}[dingli]{\hei 引理~}
```

```
\newtheorem{tuilun}[dingli]{\hei 推论~}
```

```
\newtheorem{mingti}[dingli]{\hei 命题~}
```

```
\newtheorem{lizi}{{例}}
```

```
%
```

```
%
```

### 3.3 中文字体使用范例

华文中宋、华文仿宋、**华文隶书**、**华文行楷**、**华文琥珀**、微软雅黑、方正姚体、方正舒体

### 3.4 设置不同的字体大小

初号、小初号、三号、六号、七号

### 3.5 总结

首先在 tex 文件内导入宏包 `\usepackage{xeCJK}`，然后基于下面两个步骤实现对本地系统中任意字体的使用。

1. 首先，将 tex 系统不认识的这个本地字体加入到 CJKfamily 大家庭，使其被 tex 系统识别（理解即可），操作方法如下：

```
\setCJKfamilyfont{重新命名的字体, 如 hyxjj}{系统内的某个字体名, 如  
汉仪雪君体简.TTF}
```

2. 然后, 定义新的字体调用命令。`\newcommand{\hyxjj}{\CJKfamily{hyxjj}}`

之后在 tex 正文中, 通过调用上面创立的新命令 `{\hyxjj xxx}` 实现对文本 xxx 的修饰, 另外注意大括号在两边, 命令及修饰文本在里边, 因为该命令不是带参数式的命令, 而是声明型的命令。注意 tex 系统字体的编码方式必须设定为 UTF-8 模式（这里的编码就相当于对不同的字体排个序号, 即一字一号, 否则容易出现字体乱码）, 编译时采用 X<sub>Ǝ</sub>LT<sub>Ǝ</sub>X 编译。网络中大家可以找到很多其他的使用说明教程, 通过配置宏包来支持不同字体, 具体宏包指令也与这种方法中的指令相似。对于 X<sub>Ǝ</sub>LT<sub>Ǝ</sub>X 的爱好者或者需要经常进行排版的专业人士而言, 自己编写这样一个宏包或者使用第三方宏包无疑是最好的选择。over!