

# CT<sub>E</sub>X 宏集手册

CTEX.ORG

2026/07/01 v2.6.1\*

## 简介

CT<sub>E</sub>X 宏集是面向中文排版的通用 L<sup>A</sup>T<sub>E</sub>X 排版框架, 为中文 L<sup>A</sup>T<sub>E</sub>X 文档提供了汉字输出支持、标点压缩、字体字号命令、标题文字汉化、中文版式调整、数字日期转换等支持功能, 可适应论文、报告、书籍、幻灯片等不同类型的中文文档。

CT<sub>E</sub>X 宏集支持 L<sup>A</sup>T<sub>E</sub>X、pdfL<sup>A</sup>T<sub>E</sub>X、X<sub>3</sub>L<sup>A</sup>T<sub>E</sub>X、LuaL<sup>A</sup>T<sub>E</sub>X、upL<sup>A</sup>T<sub>E</sub>X 等多种不同的编译方式, 并为它们提供了统一的界面。主要功能由宏包 `ctex` 以及中文文档类 `ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer` 实现。

## 目录

<b>第 1 节 介绍</b>	<b>2</b>	<b>第 7 节 章节标题样式设置</b>	<b>16</b>
<b>第 2 节 简明教程</b>	<b>3</b>	7.1 编号相关	17
2.1 CT <sub>E</sub> X 宏集的组成	3	7.2 格式相关	19
2.2 CT <sub>E</sub> X 宏集的安装和更新	3	7.3 间距、缩进相关	23
2.3 使用 CT <sub>E</sub> X 文档类	4	7.4 目录、附录相关	27
2.4 使用 <code>ctex</code> 宏包	5	7.5 辅助命令	28
<b>第 3 节 宏包选项与 <code>\ctexset</code> 命令</b>	<b>5</b>	7.6 示例	29
<b>第 4 节 编译方式、编码与中文字库</b>	<b>6</b>	<b>第 8 节 实用命令</b>	<b>29</b>
4.1 编译方式	6	8.1 字号与间距	29
4.2 中文编码	6	8.2 中文数字转换	31
4.3 中文字库	6	8.3 杂项	31
<b>第 5 节 排版格式设定</b>	<b>9</b>	<b>第 9 节 LuaL<sup>A</sup>T<sub>E</sub>X 下的中文支持方式</b>	<b>32</b>
5.1 文档默认字号	9	9.1 LuaL <sup>A</sup> T <sub>E</sub> X 下替代字体的设置	32
5.2 章节标题风格	10	<b>第 10 节 CT<sub>E</sub>X 宏集的配置</b>	<b>33</b>
5.3 排版方案选项	11	10.1 修改宏包默认选项	33
<b>第 6 节 文档汉化</b>	<b>13</b>	10.2 宏包载入后的配置	34
6.1 日期汉化	13	10.3 配置标题中文翻译	34
6.2 文档标题汉化	13	10.4 自定义字体集	34
6.3 页面格式设置与汉化	15	<b>第 11 节 第三方宏包兼容</b>	<b>34</b>

\*`ctex-kit rev. d66784e5`.

第 1 节 介绍	2
<b>第 12 节 对旧版本的兼容性</b>	<b>35</b>
2.1 C <sub>T</sub> E <sub>X</sub> 0.8a 及以前的版本 . . . . .	35
2.2 C <sub>T</sub> E <sub>X</sub> 0.9–C <sub>T</sub> E <sub>X</sub> 1.0d . . . . .	35
2.3 C <sub>T</sub> E <sub>X</sub> 1.02c 以后的 SVN 开发版	37
2.4 C <sub>T</sub> E <sub>X</sub> 2.2 之前的版本 . . . . .	37
2.5 C <sub>T</sub> E <sub>X</sub> 2.4.1 和 2.4.2 . . . . .	37
2.6 C <sub>T</sub> E <sub>X</sub> 2.5 之前的版本 . . . . .	38
<b>第 13 节 宏集依赖情况与手工安装方法</b>	<b>38</b>
<b>第 14 节 开发人员</b>	<b>39</b>
<b>第 15 节 代码实现</b>	<b>41</b>
<b>版本历史</b>	<b>166</b>
<b>代码索引</b>	<b>169</b>

## 第 1 节 介绍

### 历史

C<sub>T</sub>E<sub>X</sub> 宏集的源头有两个:一是王磊编写的 `cjkbook` 文档类,二是吴凌云编写的 `GB.cap`。这些工作没有经过认真系统的设计,也没有用户文档,不利于维护和改进。

2003 年,吴凌云使用 `doc` 和 `DocStrip` 重构了整个工程,并增加了许多新的功能,称为 `ctex` 宏包。2007 年,oseen 和王越在 `ctex` 宏包的基础上,增加了对 UTF-8 编码的支持,开发出了 `ctexutf8` 宏包。

2009 年 5 月,我们在 Google Code 建立了 `ctex-kit` 项目<sup>1</sup>,对 `ctex` 宏包及相关脚本进行了整合,并加入了对 X<sub>Y</sub>T<sub>E</sub>X 引擎的支持。在开发新版本时,考虑到合作开发和调试的方便,我们放弃了 `doc` 和 `DocStrip`,采取了直接编写宏包代码的方式。

2014 年 3 月,为了适应 L<sup>A</sup>T<sub>E</sub>X 的最新发展,特别是 L<sup>A</sup>T<sub>E</sub>X3 的逐渐成熟,李清用 L<sup>A</sup>T<sub>E</sub>X3 重构了整个宏包的代码,并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理,升级版本号为 2.0,并改称 C<sub>T</sub>E<sub>X</sub> 宏集。

2015 年 3 月,由于 Google Code 即将停止服务,`ctex-kit` 项目迁移至 [GitHub](#)<sup>2</sup>。

最初,Knuth 在设计开发 T<sub>E</sub>X 的时候没有考虑到多国文字支持,特别是对多字节的中日韩表意文字的支持。这使得 T<sub>E</sub>X 以至后来的 L<sup>A</sup>T<sub>E</sub>X 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后,中文用户使用 L<sup>A</sup>T<sub>E</sub>X 仍然要面对许多困难。这些困难里,以章节标题的中文化为最。由于中文和西文书写习惯的差异,用户很难使用标准文档类中的代码结构来表达中文标题。于是,用户不得不对标准文档类做较大的修改。除此之外,日期格式、首行缩进、中文字号和字距等细节问题,也需要精细的调校。我们设计 C<sub>T</sub>E<sub>X</sub> 宏集的目的之一就是解决这些 L<sup>A</sup>T<sub>E</sub>X 文档的汉化难题。

另一方面,随着 T<sub>E</sub>X 引擎和 L<sup>A</sup>T<sub>E</sub>X 宏包的不断发展,L<sup>A</sup>T<sub>E</sub>X 的中文支持方式从早期的专用系统(如 CCT)发展为适用于不同引擎的多种方式<sup>3</sup>。这些方式的适用情况和使用方式有不少细节上的差异,同时操作系统的不同、语言环境的不同等客观情况又进一步带来了更多的细节差异。我们设计 C<sub>T</sub>E<sub>X</sub> 宏集的另一个主要目的就是尽可能消除这些差异带来的影响,使用户能够以一个统一的接口来使用不同的中文支持方式,使得同一份文档能够在不同环境下交换使用。

C<sub>T</sub>E<sub>X</sub> 宏集的许多实现细节离不开热心朋友们在 [bbs.ctex.org](#) 论坛<sup>4</sup>上的讨论,在此对参与讨论的朋友们表示感谢。

<sup>1</sup><http://code.google.com/p/ctex-kit/>,该链接现已失效。

<sup>2</sup><https://github.com/CTeX-org/ctex-kit>

<sup>3</sup>比如:pdf<sub>T</sub>E<sub>X</sub> 引擎下的 CJK、zhmCJK 宏包,X<sub>Y</sub>T<sub>E</sub>X 引擎下的 xeCJK 宏包和 Lua<sub>T</sub>E<sub>X</sub> 引擎下的 LuaTe<sub>X</sub>-ja 宏包。

<sup>4</sup>2018 年,C<sub>T</sub>E<sub>X</sub> 论坛因故无限期关闭,此链接现已失效。

## 关于宏集名字的说明

CT<sub>E</sub>X 之名是英文单词 China(中国)或 Chinese(中文)的首字母“C”与“T<sub>E</sub>X”结合而成的。在纯文本环境下,该名字应写作“CT<sub>E</sub>X”。

CT<sub>E</sub>X 宏集是由 CT<sub>E</sub>X 社区发起并维护的 L<sup>A</sup>T<sub>E</sub>X 宏包和文档类的集合。社区另有发布名为 CT<sub>E</sub>X 套装的 T<sub>E</sub>X 发行版,与本文档所述的 CT<sub>E</sub>X 宏集并非是一事物。

ctex 则是本宏集中的 ctex.sty 的名字。这一完全小写的名称,在过去也被用来指代整个 CT<sub>E</sub>X 宏集,不过现在则特指 ctex.sty 这一宏包。在不引起歧义的情况下,它也可以沿用过去的习惯,代指整个宏集。

## 第 2 节 简明教程

### 2.1 CT<sub>E</sub>X 宏集的组成

为了适应用户不同的需求,我们将 CT<sub>E</sub>X 宏集的主要功能设计安排在四个中文文档类和三个宏包当中,具体的组成见表 1。

表 1 CT<sub>E</sub>X 宏集的组成

类别	文件	说明
文档类	ctexart.cls	标准文档类 <code>article</code> 的汉化版本,一般适用于短篇幅的文章
	ctexrep.cls	标准文档类 <code>report</code> 的汉化版本,一般适用于中篇幅的报告
	ctexbook.cls	标准文档类 <code>book</code> 的汉化版本,一般适用于长篇幅的书籍
	ctexbeamer.cls	文档类 <code>beamer</code> 的汉化版本,适用于幻灯片演示
宏包	ctex.sty	提供全部功能,但默认不开启章节标题设置功能,需要使用 <code>heading</code> 选项来开启
	ctexsize.sty	定义和调整中文字号,可以在 <code>ctex</code> 宏包或 CT <sub>E</sub> X 中文文档类之外单独调用
	ctexheading.sty	提供章节标题设置功能(见 7 节),可以在 <code>ctex</code> 宏包或 CT <sub>E</sub> X 中文文档类之外单独调用

### 2.2 CT<sub>E</sub>X 宏集的安装和更新

最常见的 T<sub>E</sub>X 发行版(T<sub>E</sub>X Live 和 MiK<sub>T</sub><sub>E</sub>X)已收录 CT<sub>E</sub>X 宏集及其依赖的宏包和宏集。<sup>5</sup>如果本地安装 T<sub>E</sub>X Live 或 MiK<sub>T</sub><sub>E</sub>X 不是完整版本,则可能需要通过这两个发行版提供的包管理器来安装宏包。

T<sub>E</sub>X Live 的包管理器是 tlmgr(T<sub>E</sub>X Live Manager)。用户可以在系统命令行中<sup>6</sup>执行

```
tlmgr gui
```

启动管理器的图形界面(Windows 用户也可以通过开始菜单的 TeX Live 20XX(年份) → TeX Live Manager 打开)。连接上远程仓库之后,搜索 `ctex` 即可安装。tlmgr 的图形界面使用 Perl 编写,在实践中发现容易造成系统假死。遇到这种问题的用户,也可以直接在系统命令行执行

<sup>5</sup>zhmCJK 宏包是个例外。当用户显式指定选项 `zhmap = zhmCJK` 时,CT<sub>E</sub>X 宏集依赖它。由于,它没有被 T<sub>E</sub>X Live 和 MiK<sub>T</sub><sub>E</sub>X 收录,用户可能需要遵照其说明文档自行安装。

<sup>6</sup>Windows 系统的命令行是 CMD 命令提示符,你可以使用 Win + R 组合键打开“运行”对话框,然后输入 `cmd` 确认打开命令提示符窗口。

```
tlmgr install ctex
```

来安装 C<sub>T</sub>E<sub>X</sub> 宏集<sup>7</sup>。

MiK<sub>T</sub>E<sub>X</sub> 通常会在缺失宏包时自动完成安装。如需手动安装，可以使用其管理维护工具 MiK<sub>T</sub>E<sub>X</sub> Console。用户可以打开管理器，连接上远程仓库之后，在“Package”选项卡中搜索“ctex”并安装即可。也可以使用 mpm(MiK<sub>T</sub>E<sub>X</sub> Package Manager)，在命令行执行

```
mpm --admin --install=ctex
```

来安装 C<sub>T</sub>E<sub>X</sub> 宏集。

若希望了解 C<sub>T</sub>E<sub>X</sub> 宏集具体的依赖情况或手工安装宏集的方法，请参阅第 13 节。

当我们将宏集的新版本发布于 CTAN，且为发行版的远程仓库更新后，用户就可以在本地通过包管理器获取新版本。

对于 T<sub>E</sub>X Live，可以在 tlmgr 的图形界面点击“更新全部已安装的”按钮或者在命令行执行

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 MiK<sub>T</sub>E<sub>X</sub>，在 MiK<sub>T</sub>E<sub>X</sub> Console 中找到“Updates”选项卡，检查更新后即可选择升级宏包。也可以使用 mpm，在命令行执行

```
mpm --admin --update
```

来进行更新。

## 2.3 使用 C<sub>T</sub>E<sub>X</sub> 文档类

如果用户需要在三个标准文档类或 *beamer* 的基础上添加中文及版式的支持，我们建议用户使用 C<sub>T</sub>E<sub>X</sub> 宏集提供的四个中文文档类。

C<sub>T</sub>E<sub>X</sub> 宏集提供了四个中文文档类：ctexart、ctexrep、ctexbook 和 ctexbeamer，分别对应 L<sup>A</sup>T<sub>E</sub>X 的标准文档类 article、report、book 和 beamer。使用它们的时候，需要将涉及到的所有源文件使用 UTF-8 编码保存<sup>8</sup>。

### 例 1

```
\documentclass{ctexart}
\begin{document}
中文档类测试。你需要将所有源文件保存为 UTF-8 编码。

你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译，也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。对高级用户，我们也推荐使用 upLaTeX 编译。
\end{document}
```

以下是使用 ctexbeamer 文档类编写中文演示文稿的一个示例。

<sup>7</sup>\*nix 用户可能需要超级用户权限(sudo)才能正确安装宏集。

<sup>8</sup>使用 (pdf)L<sup>A</sup>T<sub>E</sub>X 时也能够使用 GBK 编码，但不推荐。(见 4.2 节)

例 2

```

\documentclass{ctexbeamer}
\begin{document}
\begin{frame}{中文演示文档}
\begin{itemize}
\item 你需要将所有源文件保存为 UTF-8 编码
\item 你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译
\item 也可以使用 (pdf)LaTeX 编译
\item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\item 对高级用户，我们也推荐使用 upLaTeX 编译
\end{itemize}
\end{frame}
\end{document}

```

## 2.4 使用 `ctex` 宏包

用户在使用非标准文档类及 *beamer* 时，如果需要添加中文及版式的支持，则可以使用 `ctex` 宏包。

对于建立在 L<sup>A</sup>T<sub>E</sub>X 标准文档类之上开发的文档类，在使用 `ctex` 宏包时加上 `heading` 选项，可以将章节标题设置为中文风格。

例 3

```

\documentclass{ltxdoc}
\usepackage[heading = true]{ctex}
\begin{document}
\section{简介}
章节标题中文化的 \LaTeX{} 手册。
\end{document}

```

## 第 3 节 宏包选项与 `\ctexset` 命令

CT<sub>E</sub>X 宏集已经尽可能就中文的行文和版式习惯做了调整和配置，通常而言，这些配置已经够用。因此，除非必要，我们不建议普通用户修改这些默认配置。如果你认为 CT<sub>E</sub>X 宏集的默认配置还可以完善，可以在项目主页上[提交 issue](#)，向我们反映，我们会酌情在后续版本中予以改进。

不过，CT<sub>E</sub>X 宏集也提供了一系列选项。用户可以使用这些选项来控制 CT<sub>E</sub>X 宏集的行为。按形式分类，这些选项有的以传统的方式提供，有的以 `<key>=<value>` 的形式提供。按指定位置分类，这些选项又可以分为以下三类：

- 名字后带有 ☆ 号的选项，只能作为宏包/文档类选项，需要在引入宏包/文档类的时候指定；
- 名字后带有 ★ 号的选项，只能通过 CT<sub>E</sub>X 宏集提供的用户接口 `\ctexset` 来设定；
- 名字后不带有特殊符号的选项，既可以作为宏包/文档类选项，也可以通过 `\ctexset` 来设定。

后续文档将在使用说明中对某些特殊的选项加以说明。

---

`\ctexset` `\ctexset` `{(键值列表)}`

是 CT<sub>E</sub>X 宏集的通用控制命令，用来在宏包载入后控制宏包的各项功能。`\ctexset` 的参数是一个键值列表，以通用的接口完成各项设置。

`\ctexset` 的参数是一组由逗号分隔的选项列表，列表中的选项通常是一个 `<key>=<value>` 格式的定义。例如设置摘要与参考文献标题名称(6.2 节)就可以使用：

例 4

```
\ctexset{
  abstractname = {本文概要},
  bibname      = {文\quad 献}
}
```

`\ctexset` 采用  $\LaTeX$ 3 风格的键值设置, 支持不同类型的选项与层次化的选项设置, 相关示例见 7 节。

## 第 4 节 编译方式、编码与中文字库

### 4.1 编译方式

$\CTeX$  宏集会根据用户使用的编译方式<sup>9</sup>, 在底层选择不同的中文支持方式(见表 2)。

表 2  $\CTeX$  宏集的中文支持方式

编译方式	(pdf) $\LaTeX$	$X_{\LaTeX}$	Lua $\LaTeX$	up $\LaTeX$ *
支持宏包	CJK	xeCJK	LuaTeX-ja	原生

\*  $p\LaTeX$ -ng (或称  $A\LaTeX$ ) 与  $up\LaTeX$  兼容。使用  $p\LaTeX$ -ng 编译时, `ctex` 采用与  $up\LaTeX$  相同的设置。

不同的编译方式和中文支持方式会在一定程度上影响  $\CTeX$  宏集的行为, 比如宏包对文档编码、字体选择、空格、标点等的处理。具体细节将在本文档后续内容中进行阐述。

### 4.2 中文编码

**GBK** ☆ 指明编写文档时使用的编码。 $\CTeX$  宏集无法检测文档源文件的实际编码格式, 因此需要用户通过选项声明。如果没有显式指定, 则默认采用 UTF-8 编码。

**UTF8** ☆

使用  $X_{\LaTeX}$ 、Lua $\LaTeX$  或  $up\LaTeX$  编译时,  $\CTeX$  宏集强制使用 UTF-8 编码, 此时 GBK 选项无效; 使用 (pdf) $\LaTeX$  编译时,  $\CTeX$  宏集默认使用 UTF-8 编码, 但用户也可以显式声明 GBK 选项, 使  $\CTeX$  宏集按 GBK 编码处理文档。

用户需要保证编译方式、源文件编码、宏包编码选项三者一致。

我们建议编写新文档时始终使用 UTF-8 编码, 而仅把 GBK 编码留给历史遗留文档。

### 4.3 中文字库

以往, 为  $\LaTeX$  文档配置中文支持是一件相当繁琐的事情。默认情况下,  $\CTeX$  宏集能自动检测用户使用的编译方式(参见 4.1 节)和操作系统<sup>10</sup>, 选择合适的底层支持和字库, 从而简化配置过程。自动配置的情况参见表 3。

通常, 由  $\CTeX$  宏集进行的自动配置已经足够使用, 无需用户手工干预; 但是  $\CTeX$  仍然提供了一系列选项, 供在  $\CTeX$  的自动选择机制因为意外情况失效, 或者在用户有特殊需求的情况下使用。除非必要, 用户不应使用这些选项。

<sup>9</sup>  $\LaTeX$ 、pdf $\LaTeX$ 、 $X_{\LaTeX}$ 、Lua $\LaTeX$  及  $up\LaTeX$ 。

<sup>10</sup>  $\CTeX$  宏集现在能够识别 macOS 及 Windows 系统, 并将其他系统统一归为 Linux。

表 3 CT<sub>E</sub>X 宏集自动配置字体策略

	macOS Old <sup>1</sup>	macOS New <sup>2</sup>	Windows <sup>3</sup>	其他
X <sub>Y</sub> L <sub>A</sub> T <sub>E</sub> X	xeCJK 华文字库	xeCJK 华文字库 + 苹方	xeCJK 中易字库 + 微软雅黑	xeCJK Fandol 字库 <sup>4</sup>
Lua <sub>L</sub> A <sub>T</sub> E <sub>X</sub> <sup>5</sup>	LuaTeX-ja 华文字库	LuaTeX-ja 华文字库 + 苹方	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja Fandol 字库
pdf <sub>L</sub> A <sub>T</sub> E <sub>X</sub>	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑 <sup>6</sup>	不可用
L <sub>A</sub> T <sub>E</sub> X + DVIPDFM <sub>X</sub>	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑 <sup>6</sup>	CJK + zhmetrics Fandol 字库
up <sub>L</sub> A <sub>T</sub> E <sub>X</sub> + DVIPDFM <sub>X</sub>	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

<sup>1</sup> Yosemite (10.10) 及以前的 macOS 系统。

<sup>2</sup> El Capitan (10.11) 及以后的 macOS 系统。

<sup>3</sup> 仅支持 Windows Vista 及以后的 Windows 操作系统。

<sup>4</sup> 由马起园、苏杰、黄晨成等人开发的开源中文字体, 参见: <https://www.ctan.org/pkg/fandol>。

<sup>5</sup> Lua<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。

<sup>6</sup> 微软雅黑字体并不总是有效, 这和选项 zhmap 的取值有关。

`zhmap` ☆ `zhmap = <true|false|zhmCJK>`

指定字体映射机制。本选项只在使用 (pdf)<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 编译时有意义。

`true` 这是该选项的默认值。<sup>11</sup>使用 `zhmetrics` 宏包, 将 CJK 字库通过 `\special` 命令映射到 `.ttf` 文件。

`false` 使用传统的 CJK 字库 (Type 1)<sup>12</sup>。

`zhmCJK` 载入 `zhmCJK` 宏包<sup>13</sup>, 由 `zhmCJK` 宏包提供从 CJK 字库到 `.ttf` 的映射。

`fontset` `fontset = <adobe|fandol|founder|hanyi|mac|macnew|macold|ubuntu|windows|none|...>`

指定 CT<sub>E</sub>X 宏集加载的字库。

如果没有指定 `fontset` 的值, CT<sub>E</sub>X 宏集将自动检测用户使用的操作系统, 配置相应的字体 (参见表 3)。

CT<sub>E</sub>X 预定义了以下七种中文字库。

`adobe` 使用 Adobe 公司的四款中文字体, 不支持 pdf<sub>L</sub>A<sub>T</sub>E<sub>X</sub>。

`fandol` 使用 Fandol 中文字体, 不支持 pdf<sub>L</sub>A<sub>T</sub>E<sub>X</sub>。

`founder` 使用方正公司的中文字体。

`hanyi` 使用汉仪字库的中文字体。

`mac` 使用 macOS 系统下的字体, 不支持 pdf<sub>L</sub>A<sub>T</sub>E<sub>X</sub>, 根据版本分为 `macnew` 和 `macold` 两种。

`macnew` 使用 El Capitan 或之后的多字重华文字体和苹方字体。从 macOS 15 (Sequoia) 开始, 部分字体 (苹方、楷体、仿宋、隶书、圆体) 转为 `downloadable`, CT<sub>E</sub>X 会自动检测可用性: 可用时使用, 不可用时静默跳过或回退。若需要完整的字体集, 请在系统字体册中手动下载相关字体。

注意: 上述自动检测仅对 `macnew` 字体集中的默认字体生效。如果需要在 Lua<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 下通过 `\setCJKmainfont` 等命令自定义使用 macOS 的 `downloadable` 字体, 需要配置 `OSFONTDIR` 环境

<sup>11</sup> 对于以键值对形式提供的选项, 文档以**粗体**来表示其默认值。下同, 不再额外解释。

<sup>12</sup> 使用 (pdf)<sub>L</sub>A<sub>T</sub>E<sub>X</sub> 编译时, 如果需要使用自定义的字体映射文件 (比如需要使用 L<sub>A</sub>T<sub>E</sub>X + Dvips 编译), 或者希望使用 Type1 字库, 请禁用本选项。为此, 你可能需要安装 CJK 字体。参考 `zhmetrics` 宏包提供的脚本 `CTeXFonts.lua`。

<sup>13</sup> `zhmCJK` 宏包基于 `zhmetrics` 和 `CJK` 宏包, 提供与 `xeCJK` 宏包类似的用户接口。

变量, 将字体目录加入搜索路径。例如, 在 shell 配置文件中添加:

例 5

```
export OSFONTDIR="/System/Library/Fonts;/Library/Fonts;~/Library/Fonts"
```

配置后需运行 `luaotfload-tool --update` 重建字体数据库。Xe<sub>La</sub>TeX 不受此限制, 因为 Xe<sub>La</sub>TeX 自身可通过系统接口发现字体。

- macold 使用 Yosemite 或之前的华文字体。
- ubuntu 使用 Ubuntu 系统下的思源宋体、思源黑体和 TeX 发行版自带的文鼎楷体, 不支持 pdf<sub>La</sub>TeX。
- windows 使用 Windows 系统下的中易字体和微软雅黑字体。当使用 (pdf)<sub>La</sub>TeX 编译时, 微软雅黑仅在以下两种情形有效: 安装有 zhmCJK 宏包且选项 `zhmap=zhmCJK` 时, 或者安装有微软雅黑的 Type1 字体且选项 `zhmap=false` 时。

如果不想使用 C<sub>T</sub>EX 预定义的中文字库, 可以设置 `fontset` 为下述值之一。

- none 不配置中文字体, 需要用户自己配置。
- (name) 这里 (name) 为自定义的名字。C<sub>T</sub>EX 宏集将载入名为 `ctex-fontset-(name).def` 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录树下合适位置建立一个名为 `ctex-fontset-(name).def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=(name)` 选项来调用它。字体配置文件的具体写法可以参考 C<sub>T</sub>EX 宏集 `fontset` 目录下的字体配置文件。

注意: 如果希望使用 `\ctexset` 在导言区指定字库, 则需要先在宏包/文档类选项中指定 `fontset = none` (这会禁用 C<sub>T</sub>EX 宏集的操作系统检测功能和自动设定字库功能)。例如:

例 6

```
\documentclass[fontset = none]{ctexart}
\ctexset{fontset = founder}
\begin{document}
在文档类选项中声明 \verb|fontset = none|, 随后在导言区用 \verb|\ctexset|
指定字体。
\end{document}
```

C<sub>T</sub>EX 宏集预定义的中文字库还定义了一些字体命令。除了在 ubuntu 字库中没有 `\fangsong` 的定义外, 所有字库都有以下四个字体命令:

- `\songti` 宋体, CJK 等价命令 `\CJKfamily{zhsong}`。
- `\heiti` 黑体, CJK 等价命令 `\CJKfamily{zhhei}`。
- `\fangsong` 仿宋, CJK 等价命令 `\CJKfamily{zhfs}`。
- `\kaishu` 楷书, CJK 等价命令 `\CJKfamily{zhkai}`。

在 windows、founder 和 macnew 字库中, 额外定义了 `\lishu` 和 `\youyuan`:

- `\lishu` 隶书, CJK 等价命令 `\CJKfamily{zhli}`。
- `\youyuan` 圆体, CJK 等价命令 `\CJKfamily{zhyou}`。

在 windows 字库中还定义了 `\yahei`。出于兼容性的考虑, `\yahei` 命令在 macnew 字库中也有定义, 但实际调用苹方黑体:

- `\yahei` 微软雅黑, CJK 等价命令 `\CJKfamily{zhyahei}`。

在 macnew 字库中, 还定义了 `\pingfang`:

- `\pingfang` 苹方黑体, CJK 等价命令 `\CJKfamily{zhpff}`。在 macOS 15 及之后, 苹方字体可能尚未下载, 此时该命令将回退为 `\heiti` 的别名。

## 第 5 节 排版格式设定

### 5.1 文档默认字号

experiment/font-size-system ☆ experiment/font-size-system = <word|letterpress|<自定义名称>>

New: 2026-05-04

选择字号系统。本选项位于 `experiment` 命名空间下, 表明其实验性质。本选项只能作为文档类或宏包选项使用, 不能在导言区通过 `\ctexset` 设置。例如:

```
\documentclass[experiment/font-size-system=letterpress]{ctexart}
```

CT<sub>E</sub>X 提供两种预设字号系统:

`word` 默认值。源自 Microsoft Word / Adobe 等桌面排版软件的字号定义, 亦与 GB/T 40070—2021 国家标准一致。这是 CT<sub>E</sub>X 历史以来使用的字号体系。

`letterpress` 金属活字排印字号体系的一种, 基于严格的倍数关系。初号/二号/五号/七号形成  $\times 2$  等比序列 (42 : 21 : 10.5 : 5.25), 一号/四号形成  $\times 2$  序列 (28 : 14), 三号/六号/八号形成  $\times 2$  序列 (16 : 8 : 4)。历史上金属活字字号曾存在多种命名与点数约定, 本预设仅为其中一种, 并非唯一定义。

两种字号系统仅在以下六个字号上存在差异:

	word (bp)	letterpress (bp)	倍数关系
一号 (1)	26	28	四号 $\times 2$
二号 (2)	22	21	五号 $\times 2$
六号 (6)	7.5	8	三号 $\div 2$
小六 (-6)	6.5	7.5	小三 $\div 2$
七号 (7)	5.5	5.25	二号 $\div 4$
八号 (8)	5	4	六号 $\div 2$

两种字号系统的其余十个字号完全相同, 且行距公式(字号的 1.2 倍)不变。

用户也可以定义自己的字号系统。将字号数据保存在 `ctex-fontsize-⟨name⟩.def` 文件中, 放到工作目录或 T<sub>E</sub>X 目录树 (TDS) 中, 然后指定 `experiment/font-size-system=⟨name⟩` 即可。自定义文件应使用 `\ctex_save_font_size:nn` 为每个字号注册数据。例如:

```
% 文件: ctex-fontsize-custom.def
\clist_map_inline:nn
{
  { 8 } { 5 bp } ,
  { 7 } { 5.5 bp } ,
  { -6 } { 6.5 bp } ,
  { 6 } { 7.5 bp } ,
  { -5 } { 9 bp } ,
  { 5 } { 10.5 bp } ,
  { -4 } { 12 bp } ,
  { 4 } { 14 bp } ,
  { -3 } { 15 bp } ,
  { 3 } { 16 bp } ,
  { -2 } { 18 bp } ,
  { 2 } { 22 bp } ,
  { -1 } { 24 bp } ,
  { 1 } { 26 bp } ,
  { -0 } { 36 bp } ,
  { 0 } { 42 bp }
}
{ \ctex_save_font_size:nn #1 }
```

---

`zihao` ☆ `zihao = (-4|5|false)`

将文章默认字号 (`\normalsize`) 设置为小四号字或五号字, 具体情况见表 4。 `false` 禁用本功能。本选项可以用于四个 C<sub>T</sub>E<sub>X</sub> 文档类和 `ctex` 宏包, 也可以用于 `ctexsize` 宏包。

该选项的默认值与 `scheme` 的取值有关。当 `scheme = chinese` 时, 对标准文档类默认值为 5, 即设置 `\normalsize` 为五号字; 对 `beamer` 则为 `false`, 使用文档类原有的设置。当 `scheme = plain` 时, 该选项不设默认值, 沿用标准文档类或 `beamer` 的原有设置。

表 4 标准字体命令与字号的对应

字体命令	zihao = 5		zihao = -4		10pt	11pt	12pt
	字号	bp	字号	bp	pt	pt	pt
<code>\tiny</code>	七号	5.5	小六	6.5	5	6	6
<code>\scriptsize</code>	小六	6.5	六号	7.5	7	8	8
<code>\footnotesize</code>	六号	7.5	小五	9	8	9	10
<code>\small</code>	小五	9	五号	10.5	9	10	11
<code>\normalsize</code>	五号	10.5	小四	12	10	11	12
<code>\large</code>	小四	12	小三	15	12	12	14
<code>\Large</code>	小三	15	小二	18	14	14	17
<code>\LARGE</code>	小二	18	二号	22	17	17	20
<code>\huge</code>	二号	22	小一	24	20	20	25
<code>\Huge</code>	一号	26	一号	26	25	25	25

表 4 中字体命令对应的字号大小(bp)基于默认的 word 字号系统。当启用 `experiment/font-size-scheme = letterpress` 时, 一号、二号、六号、小六、七号、八号这六个字号的 bp 数值会发生变化, 其余字号保持不变。具体差异见 5.1 节。

---

10pt ☆ C<sub>T</sub>E<sub>X</sub> 文档类是在 L<sub>A</sub>T<sub>E</sub>X 标准文档类之上开发的。因此, 除了可以使用 C<sub>T</sub>E<sub>X</sub> 宏包定义的字号选项之外, 还可以使用标准文档类的同类选项 (10pt、11pt 和 12pt)。在使用这些来自标准文档类的选项的时候, C<sub>T</sub>E<sub>X</sub> 文档类的字号选项会被抑制。亦即, 在 `zihao` 选项之后设置 10pt 选项, `zihao` 选项将不再起作用。

标准文档类的其他选项在 C<sub>T</sub>E<sub>X</sub> 文档类中依旧有效。例如, 设置纸张大小和方向的 `a4paper` 和 `landscape`, 设置单双面的 `oneside` 和 `twoside` 等。C<sub>T</sub>E<sub>X</sub> 会将这些选项传给标准文档类<sup>14</sup>。

## 5.2 章节标题风格

---

`heading` ☆ `heading = (true|false)`

本选项只能在调用 `ctex.sty` 时作为宏包选项使用。

C<sub>T</sub>E<sub>X</sub> 宏集提供了一套用于修改文档章节标题格式的接口。该选项用于选择是否启用该功能。详细的设置方法请参见第 6.3 节和第 7 节。

C<sub>T</sub>E<sub>X</sub> 宏集提供的四个文档类总是启用该功能。如果在 `ctex.sty` 下启用该选项, 将会检查当前是否使用 L<sub>A</sub>T<sub>E</sub>X 标准文档类。若然, 则该选项将会使得 `ctex.sty` 宏包的行为和 C<sub>T</sub>E<sub>X</sub> 宏集提供的四个中文文档类完全一致; 若不然, 则会根据 `\chapter` 是否有定义来使用 `ctexbook` 或者 `ctexart` 的标题设置。

<sup>14</sup>事实上, L<sub>A</sub>T<sub>E</sub>X 在文档类中的选项是全局设定的, 除了对使用的文档类有影响外, 也可能影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中, 那么该选项将会被自动激活。

---

`sub3section` ☆ 修改 `\paragraph` 和 `\subparagraph` 的格式。

`sub4section` ☆ 默认情况下, `\paragraph` 和 `\subparagraph` 会将标题与随后的正文排版在同一个段落。启用 `sub3section` 会将 `\paragraph` 的格式修改为类似 `\section` 的格式, 并将 `\subparagraph` 的格式修改为原本 `\paragraph` 的格式。启用 `sub4section` 会将 `\paragraph` 和 `\subparagraph` 的格式都修改为类似 `\section` 的格式。

启用该选项通常还需要将计数器 `secnumdepth` 的值为设置为 4 或 5。

具体格式可参考 7.3 小节中的 `runin` 和 `afterskip` 选项。

注意, 上述两个选项只有在非 `beamer` 文档类下 `heading` 选项启用的时候才有意义。亦即, 只有在使用除了 `ctexbeamer` 的三个  $\text{\TeX}$  文档类或启用了 `heading` 的 `ctex.sty` 的时候才有意义。

### 5.3 排版方案选项

---

`scheme` ☆ `scheme = (chinese|plain)`

选择文章的排版方案, 预设有 `chinese` 和 `plain` 两种方案。

`chinese` 对 `beamer` 以外的文档类, 调整默认字号为五号字, 并调整行距为 1.3; 汉化文档中的标题名字 (如“图”、“表”、“目录”和“参考文献”等, 见 6.2 节); 在 `heading = true` 的情况下<sup>15</sup>(5.2 节), 还会将章节标题的风格修改为中文样式 (见 7 节)。

当关闭 `heading` 选项的 `ctex` 宏包与标准文档类或其衍生文档类联用时, 会载入 `indentfirst` 宏包, 以实现章节标题后的段首缩进。

`plain` 不调整默认字号和行距, 不会汉化文档中的标题名字, 也不会将章节标题风格修改为中文样式, 同时不会调整 `\pagestyle`, 并禁用 `autoindent` 选项。事实上, 此时的  $\text{\TeX}$  宏集只提供了中文支持功能, 而不对文章版式进行任何修改。

---

`punct` `punct = (quanjiao|banjiao|kaiming|CCT|plain)`

设置标点处理格式。预定义好的格式有:

`quanjiao` 全角式: 所有标点占一个汉字宽度, 相邻两个标点占 1.5 汉字宽度;

`banjiao` 半角式: 所有标点占半个汉字宽度;

`kaiming` 开明式: 句末点号<sup>16</sup>占一个汉字宽度, 标号和句内点号占半个汉字宽度;

`CCT` CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

`plain` 原样 (不调整标点间距)。

---

`space` `space = (true|false|auto)`

是否在生成的 PDF 中保留汉字后面的空格。该选项仅在使用  $\text{\TeX}$ /(pdf) $\text{\TeX}$  编译时有效。

`true` 总是保留汉字后的空格。此时, 用户需要自行在行尾加上 `%` 处理换行产生的空格<sup>17</sup>。

`false` 使用 (pdf) $\text{\TeX}$  编译时: 总是忽略掉汉字后面的空格, 不论汉字后是什么; 使用  $\text{\TeX}$  编译时, 等同于 `auto` 的效果。不建议使用该选项。

`auto` 根据空格后面的情况决定是否保留: 如果空格后面是汉字, 则忽略该空格, 否则保留。

例如, 使用

<sup>15</sup>使用  $\text{\TeX}$  文档类, 或者使用 `ctex` 宏包且开启该选项时。

<sup>16</sup>标点符号分为标号与点号。点号分为两类, 一共七种: 句末点号有句号、问号和叹号; 句内点号有逗号、顿号、冒号和分号。

<sup>17</sup> $\text{\TeX}$  将单个换行视作一个空格。

## 例 7

```
\ctexset{space=true}
汉字 分词
技术 English
```

将得到“汉字 分词 技术 English”；使用

## 例 8

```
\ctexset{space=auto}
汉字 分词
技术 English
```

则会得到“汉字分词技术 English”。

使用 Lua $\LaTeX$  及 up $\LaTeX$  编译的时候, 该选项无效: 汉字间的空格以及汉字与西文字符之间的空格总是有效, 不会被忽略, 但可以自动忽略掉由换行产生的空格。

---

**linespread** ☆ `linespread = <数值>`

接受一个浮点数值, 设置行距倍数。本选项的初始值与 `scheme` 有关。

`scheme = chinese` 对标准文档类初始值为 1.3, 即 1.3 倍行距。此时, 相邻两行的基线 (`\baselineskip`) 距离为  $1.3 \times 1.2 = 1.56$  倍字体高度。对 `beamer` 不改变行距, 即使用默认的单倍行距。

`scheme = plain` CT $\TeX$  宏集默认不调整行距倍数, 文档中的行距由所选文档类和其他宏包或用户设置决定。

---

**autoindent** `autoindent = <true|false|数值|带单位的数值>`

在字体大小发生变化时, 是否自动调整段首缩进 (`\parindent`) 的大小。

<数值或带单位的数值> 用于设置段首缩进的长度。如果不带单位, 则默认单位是单个汉字字宽 `\ccwd`; 如果带单位, 则使用该单位。若要显式使用 `\ccwd` 为单位, 则必须在导言区进行设置。

`true` 等价于设置 `autoindent = 2` 或在导言区设置 `autoindent = 2\ccwd`。

`false` 禁用自动调整功能, 可以设置固定长度的段首缩进。如设置每段缩进 40 点:

## 例 9

```
\ctexset{autoindent=false}
\setlength\parindent{40pt}
```

---

**linestretch** ☆ `linestretch = <数值或长度>`

`linestretch` 是一个比较特殊的选项, 它用来设置汉字之间弹性间距的弹性程度。如果有单位, 则可以在选项中直接写; 如果是数字, 单位则是汉字宽度 `\ccwd` 的倍数。

如果行宽不是汉字宽度的整数倍, 为了让段落左右两端对齐, 自然就要求伸展汉字之间的间距, 而 `linestretch` 选项就是设置每行总的允许伸行量。初始值是允许每行伸行一个汉字的宽度 `\ccwd`, 并且此宽度能根据字号变化动态调整。

过小的 `linestretch` 可能导致段落文字右侧可能参差不齐; 较大的 `linestretch` 选项则可以帮助拥有较长不可断行内容的复杂段落方便地断行, 而不会产生大量编译警告; 但很大的 `linestretch` 则会掩盖段落不良断行产生的坏盒子警告。

如果将 `linestretch` 选项的值设置为 `\maxdimen`, 则可以禁止按字号自动修改每行的允许伸长度。此时汉字间的弹性间距则固定为 `\baselineskip` 的 0.08 倍。

---

```
experiment/CJKecglue experiment/CJKecglue = {<弹性长度>}
```

---

New: 2026-05-03

设置 CJK 文字与西文之间的间距（汉英文间胶）。这是一个实验性选项，目前仅在  $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\text{L}^{\text{u}}\text{a}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  和  $\text{u}\text{p}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  下有效，使用  $\text{p}\text{d}\text{f}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  编译时会产生警告。

在  $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  下，此选项等价于设置  $\backslash\text{x}\text{e}\text{C}\text{J}\text{K}\text{s}\text{e}\text{t}\text{u}\text{p}\{\text{C}\text{J}\text{K}\text{e}\text{c}\text{g}\text{l}\text{u}\text{e}=\{\backslash\text{h}\text{s}\text{k}\text{i}\text{p}\ \dots\}\}$ ；在  $\text{L}^{\text{u}}\text{a}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  和  $\text{u}\text{p}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  下，此选项设置  $\text{x}\text{k}\text{a}\text{n}\text{j}\text{i}\text{s}\text{k}\text{i}\text{p}$ 。例如：

## 例 10

```
\ctexset{experiment/CJKecglue = 0.25em plus 0.15em minus 0.05em}
```

请注意，由于  $\text{p}\text{d}\text{f}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  下的 CJK 宏包没有独立的汉英文间胶概念，此选项无法在所有引擎下提供一致的行为。因此，除非确定文档仅会在  $X_{\text{q}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  或  $\text{L}^{\text{u}}\text{a}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  下编译，否则不建议使用此选项。

## 第 6 节 文档汉化

### 6.1 日期汉化

$\text{C}\text{T}_{\text{E}}\text{X}$  宏包对显示当前日期的  $\backslash\text{t}\text{o}\text{d}\text{a}\text{y}$  命令进行了汉化，使之以中文的方式显示今天的日期。如本文档编译时的日期是“2026 年 7 月 2 日”。

---

```
today * today = <small|big|old>
```

---

该选项用来控制  $\backslash\text{t}\text{o}\text{d}\text{a}\text{y}$  命令的输出格式：

`small` 效果为“2026 年 7 月 2 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇二六年七月二日”。使用全汉字的日期格式。

`old` 效果为“July 2, 2026”。使用文档原来的(英文)日期格式。

设置日期格式使用  $\backslash\text{c}\text{t}\text{e}\text{x}\text{s}\text{e}\text{t}$  命令完成，例如设置全汉字的日期格式：

## 例 11

```
\ctexset{today=big}
```

$\text{C}\text{T}_{\text{E}}\text{X}$  宏包的中文日期功能实际上是调用  $\text{z}\text{h}\text{n}\text{u}\text{m}\text{b}\text{e}\text{r}$  宏包完成的。如果需要更多有关日期、时间的命令和更复杂的设置，可以查阅  $\text{z}\text{h}\text{n}\text{u}\text{m}\text{b}\text{e}\text{r}$  宏包的文档。

### 6.2 文档标题汉化

这里主要介绍由 `scheme` 选项(5.3 节)控制的文档标题汉化功能。

设置文档标题名的示例可见例 4。下面的选项(如 `contentsname`)主要用来重新定义与选项同名的宏(如  $\backslash\text{c}\text{o}\text{n}\text{t}\text{e}\text{n}\text{t}\text{s}\text{n}\text{a}\text{m}\text{e}$ )的定义。

---

```
contentsname * contentsname = <名字>
```

---

设置目录标题名  $\backslash\text{c}\text{o}\text{n}\text{t}\text{e}\text{n}\text{t}\text{s}\text{n}\text{a}\text{m}\text{e}$ 。中文默认为“目录”。

---

```
listfigurename * listfigurename = <名字>
```

---

设置插图目录标题名  $\backslash\text{l}\text{i}\text{s}\text{t}\text{f}\text{i}\text{g}\text{u}\text{r}\text{e}\text{n}\text{a}\text{m}\text{e}$ 。中文默认为“插图”。

---

`listtablename` \* `listtablename = <名字>`

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

---

`figurename` \* `figurename = <名字>`

设置图片环境标题名 `\figurename`。中文默认为“图”。

---

`tablename` \* `tablename = <名字>`

设置表格环境标题名 `\tablename`。中文默认为“表”。

---

`abstractname` \* `abstractname = <名字>`

设置摘要 `abstract` 环境标题名 `\abstractname`。中文默认为“摘要”。注意 `book` 类没有摘要，该选项无效。

---

`indexname` \* `indexname = <名字>`

设置索引标题名 `\indexname`。中文默认为“索引”。

---

`appendixname` \* `appendixname = <名字>`

设置附录标题名 `\appendixname`。中文默认为“附录”。

---

`bibname` \* `bibname = <名字>`

设置参考文献标题名。中文默认为“参考文献”。

在标准文档类中 `article` 的参考文献名使用宏 `\refname`，而 `book` 和 `report` 使用宏 `\bibname`。本选项会根据标准文档类的不同，自动设定 `\refname` 或是 `\bibname`。因此，对于标准文档类及对应的 `CTEX` 文档类可以统一地使用 `bibname` 选项来控制参考文献标题名。

对于 `beamer` 及对应的 `ctexbeamer` 来说，它们同时具有宏 `\bibname` 和宏 `\refname`。本选项仅控制其中的 `\bibname`；`\refname` 则交由 `refname` 选项控制。

---

`proofname` \* `proofname = <名字>`

设置证明环境的名称 `\proofname`。中文默认为“证明”。

如果使用 `ctexbeamer` 文档类或者在 `beamer` 文档类下使用 `ctex` 包，还会汉化常用定理类环境的诸如“定义”、“定理”和“引理”等名称。此时，还有下列三个选项。

---

`refname` \* `refname = <名字>`

设置参考文献标题名 `\refname`。中文默认为“参考文献”。

注意，三个标准文档类（及相应的 `CTEX` 文档类）的参考文献标题名由 `bibname` 选项统一设置，本选项仅适用于 `beamer` 及其对应的 `ctexbeamer`。在三个标准文档类（及相应的 `CTEX` 文档类）中使用 `refname` 选项会报错。

---

`algorithmname` \* `algorithmname = <名字>`

设置 `beamer` 中的算法环境标题名 `\algorithmname`。中文默认为“算法”。

---

`continuation` \* `continuation = <名字>`

设置 `beamer` 可断页的帧在续页标题中的延续标识 `\insertcontinuationtext`。中文默认为“(续)”。

### 6.3 页面格式设置与汉化

页面格式设置与汉化的功能(及章节标题样式设置功能, 见第 7 节)由 `ctexheading` 宏包完成。加载该宏包时, 或者使用  $\text{CT}_E\text{X}$  文档类时, 或者是使用 `ctex` 宏包并设定选项 `heading = true` 时, 相关功能被激活。此时, 整个文档的页面格式(page style)被设定为 `headings`, 即相当于设置了

```
\pagestyle{headings}
```

在页眉中显示当前章节的编号与标题。

同时,  $\text{CT}_E\text{X}$  宏包也会对默认的 `headings` 页面格式进行修改, 使之调用 `\CTEXthechapter`、`\CTEXthesection` 等宏来正确显示中文的章节编号。

$\text{CT}_E\text{X}$  宏包的默认页面格式设置是经过汉化的 `headings`, 其基本效果如本文档所示, 只在页眉一侧显示章节编号和标题, 另一侧显示页码。

更复杂的页面格式可以通过调用 `fancyhdr`、`titleps` 等宏包来设置。 $\text{CT}_E\text{X}$  宏包同时也为这些自定义页面格式的包提供了以下宏供使用:

- `\CTEXthechapter`、`\CTEXthesection` 等章节编号(见 7.5 小节)。它们用来代替英文文档类中的 `\thechapter`、`\thesection` 等宏。
- `\leftmark`、`\rightmark`, 它们是在使用章节标题命令后, 自动设置的宏。它们实际是在与章节标题命令对应的标记命令 `\chaptermark`、`\sectionmark` 中调用 `\markright` 或 `\markboth` 生成的。

有关  $\text{L}_A\text{T}_E\text{X}$  页面标记的含义与使用细节, 已经超出了本文档讨论的范围。可以参考 [1, Chapter 23]、[2, §4.3, §4.4] 等书籍。

这里举一个例子, 说明通过重定义 `\sectionmark`, 在 `ctexart` 文档类中的标准 `headings` 页面格式下控制页眉的方式:

#### 例 12

```
\documentclass{ctexart}
\pagestyle{headings}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\renewcommand\sectionmark[1]{%
  \markright{\CTEXifname{\CTEXthesection--}{#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

在上例中, 我们设置了页眉的形式是用破折号分开的节编号与节标题, 即“第 1 节——天地玄黄”、“第 2 节——宇宙洪荒”。

$\text{CT}_E\text{X}$  宏包已经对 `fancyhdr` 宏包进行了补丁, 载入 `fancyhdr` 后, 其 `fancy` 页面格式将使用 `\CTEXthechapter` 等宏显示中文章节编号。

关于 `fancyhdr` 的具体用法可以参见其宏包手册。通常也只要像在标准的英文文档类中使用 `fancyhdr` 一样定义页眉页脚格式即可, 并不需要额外的定义。

下面我则给出一个与前例类似而稍复杂的例子，展示如何在文档中设置页眉内容与页眉的格式。

例 13

```
\documentclass{ctexart}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\usepackage{fancyhdr}
\fancyhf{}
\lhead{\textnormal{\kaishu\rightmark}}
\rhead{--\ thepage\ --}
\pagestyle{fancy}
% \sectionmark 的重定义需要在 \pagestyle 之后生效
\renewcommand\sectionmark[1]{%
  \markright{\CTEXifname{\CTEXthesection--}{\#1}}
}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}
```

本例的页眉效果大致如下(有页眉线):

第 1 节——天地玄黄

- 1 -

## 第 7 节 章节标题样式设置

CT<sub>E</sub>X 宏集对 L<sub>A</sub>T<sub>E</sub>X 的标准文档类 (article、report、book) 和 beamer 进行了章节标题样式设置功能的扩充。章节标题样式设置功能(及页面格式设置与汉化功能, 见第 6.3 节)由 ctexheading 宏包完成。加载该宏包时, 或者使用 CT<sub>E</sub>X 文档类时, 或者是使用 ctex 宏包并设定选项 heading = true 时, 相关功能被激活。其中, 独立使用 ctexheading 宏包时, 本节介绍各选项的默认值与指定 scheme = plain 时相同。

本节涉及的所有选项均需使用 \ctexset 命令设置。

章节标题的样式选项是分层设置的。顶层的选项是章节标题名称(例如 section), 次一层的选项是章节标题的样式(例如 nameformat)。章节标题名称包括 part, chapter, section, subsection, subsubsection, paragraph, subparagraph。可用的样式选项包括:

- 编号相关(7.1 小节): numbering, name, number
- 格式相关(7.2 小节): format, nameformat, numberformat, titleformat, aftername, aftertitle, pagestyle
- 间距、缩进相关(7.3 小节): runin, hang, indent, beforekip, afterskip, fixskip, break, afterindent
- 目录、附录相关(7.4 小节): tocline, lofskip, lotskip, appendix/numbering, appendix/name, appendix/number

注意, 对 article 及其衍生的 ctexart 等文档类, 没有 chapter 级别的标题; 而对于 beamer 文档类, 这些选项控制的是由 \partpage, \sectionpage 和 \subsectionpage 产生的标题样式, 此时只有 part, section 和 subsection 这三层级别, 并且 runin, afterindent, fixskip, hang, break 和 tocline 这六个选项无效。

多层选项之间用斜线分开, 例如, `part/name` 选项设置 `\part` 标题的在数字前后的名称, 而 `section/number` 选项设置 `\section` 标题的数字类型。

使用 `\ctexset` 设置多级选项时, 还可以在同一个上级选项下设置多个下级选项。例如, 同时设置 `part` 一级标题的 `pagestyle` 选项, `chapter` 一级标题的 `format` 与 `pagestyle` 选项和 `section` 一级标题的 `name` 与 `number` 选项:

例 14

```
\ctexset {
  part/pagestyle = empty,
  chapter = {
    format      = \raggedright,
    pagestyle   = empty,
  },
  section = {
    name        = {第,节},
    number      = \chinese{section},
  }
}
```

## 7.1 编号相关

---

<code>part/numbering</code>	* <code>numbering = true false</code>
<code>chapter/numbering</code>	*
<code>section/numbering</code>	* 控制是否对不带星号的章节标题进行编号。各级标题的默认值均为 <code>true</code> 。
<code>subsection/numbering</code>	*
<code>subsubsection/numbering</code>	*
<code>paragraph/numbering</code>	*
<code>subparagraph/numbering</code>	*

---

$\LaTeX$  标准的章节标题命令(如 `\section`)大体上完成四項工作: 输出标题内容、对标题编号(计数器增加 1)、将标题列入目录(若调用了 `hyperref` 宏包还会添加 PDF 书签)、更新页眉页脚标记。带星号的章节标题命令(如 `\section*`)只简单地输出章节标题内容, 但不标题编号, 不将标题列入目录或 PDF 书签, 也不写入页眉页脚标记。与之不同的是, 本选项仅仅是是否对不带星号的章节标题进行编号。因此, 当设置本选项为 `false` 时, 除了不对标题编号以外, 其余功能与正常标题一致: 可以编入目录, 并生成正确的 `hyperref` 目录超链接位置和页眉页脚标记。例如:

例 15

```
\documentclass{ctexbook}
\begin{document}
\tableofcontents
\chapter{A}
\chapter*{B}
\ctexset{chapter/numbering=false}
\chapter{C}
\end{document}
```

三章的标题分别为“第一章 A”、“B”和“C”, 但在目录中则只出现“第一章 A”和“C”。

注意, 章节标题是否编号还要受到  $\LaTeX$  计数器 `secnumdepth` 的控制(可通过以下介绍的 `secnumdepth` 选项设置)。例如, 对于 `section` 而言, 其深度为 1。因此, `section` 会被编号, 当且仅当 `secnumdepth` 不小于 1, 并且 `section/numbering` 为 `true`, 并且使用不带星号的章节标题命令(即 `\section`)。

---

`secnumdepth` \* `secnumdepth = <整数或章节名称>`

---

New: 2020-05-06

设置对章节标题进行编号的层次数。 `secnumdepth` 的值可以是一个整数, 也可以是 `part`, `chapter` 等名称。层次数与名称的对应关系见表 5。章节层次的默认设置见表 6。

本选项对 `beamer/ctexbeamer` 文档类无效。

表5 章节层次

层次	名称	注
-1	part	book/report 类
0	chapter	book/report 类
0	part	article 类
1	section	
2	subsection	
3	subsubsection	
4	paragraph	
5	subparagraph	

表6 章节层次的默认设置

文档类	secnumdepth	tocdepth
article	3 (subsubsection)	3 (subsubsection)
book/report	2 (subsection)	2 (subsection)
beamer	无效	3 (subsubsection)

如果没有特别说明, 以下将用“...”代表各级章节标题名。

`.../name` \* `name = {<前名字>,<后名字>}`  
`name = {<前名字>}`

设置章节的名字。所谓“章节的名字”, 可以分为前后两部分, 即章节编号前后的词语, 两个词之间用一个半角逗号分开; 也可以只有一部分, 表示只有章节编号之前的名字。例如:

例16

```
\ctexset{
  chapter/name = {第,章},
  section/name = {\S},
}
```

会使得 `\chapter` 标题使用形如“第一章”的名字, 而 `\section` 标题则使用形如“§1”的名字。该选项的默认设置见表7。

表7 name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第,部分}	{\partname\space}	原 <code>\partname</code> 为 Part
chapter	{第,章}	{\chaptername\space}	原 <code>\chaptername</code> 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 <code>\sectionname</code> 为 <code>\translate{Section}</code>
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 <code>\subsectionname</code> 为 <code>\translate{Subsection}</code>
subsection	同右	{}	
subsubsection	同右	{}	
paragraph	同右	{}	
subparagraph	同右	{}	

---

.../number \* number = {<数字输出命令>}

---

设置章节编号的数字输出格式。<数字输出命令>通常是对应章节编号计数器的输出命令，如 `\thesection` 或 `\chinese{chapter}` 之类。例如：

例 17

```
\ctexset{
  section/number = \Roman{section}
}
```

将会使 `\section` 的编号变为大写罗马数字(如 I、II 等)。

`number` 选项定义的同时将控制对章节计数器的交叉引用。在引用计数器时，记录在 L<sup>A</sup>T<sub>E</sub>X 辅助文件中的是 `number` 选项的定义。

但是，`number` 选项不会影响计数器本身的输出。即设置 `section/number` 不会影响 `\thesection` 的定义(但该选项会影响 `\CTEXthesection` 的定义，见后)。该选项的默认设置见表 8。

表 8 number 选项的默认设置

标题名	scheme = chinese	scheme = plain	原\the<标题>等价定义
part (beamer)	<code>\chinese{part}</code>	<code>\insertromanpartnumber</code>	意义为 <code>\Roman{part}</code>
part	<code>\chinese{part}</code>	<code>\thepart</code>	<code>\Roman{part}</code>
chapter	<code>\chinese{chapter}</code>	<code>\thechapter</code>	<code>\arabic{chapter}</code>
section (beamer)	同右	<code>\insertsectionnumber</code>	意义为 <code>\arabic{section}</code>
section	同右	<code>\thesection</code>	<code>\arabic{section}</code>
subsection (beamer)	<code>\arabic{section}.</code> <code>\arabic{subsection}</code>	<code>\insertsubsectionnumber</code>	意义为 <code>\arabic{subsection}</code>
subsection	同右	<code>\thesubsection</code>	<code>\thesection.\arabic{subsection}</code>
subsubsection	同右	<code>\thesubsubsection</code>	<code>\thesubsection.\arabic{subsubsection}</code>
paragraph	同右	<code>\theparagraph</code>	<code>\thesubsubsection.\arabic{paragraph}</code>
subparagraph	同右	<code>\thesubparagraph</code>	<code>\theparagraph.\arabic{subparagraph}</code>

## 7.2 格式相关

CT<sub>E</sub>X 宏集提供了 `numberformat`, `nameformat`, `titleformat`, `format` 这几个选项用来控制章节标题的格式。它们的作用范围如图 1 所示。具体用法见下文。

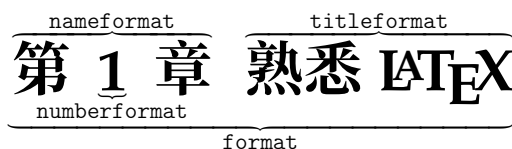


图 1 `numberformat`, `nameformat`, `titleformat`, `format` 几个选项的作用范围示意

---

```
.../format * format = {<格式命令>}
.../format+ * format+= {<格式命令>}
```

---

Updated: 2020-04-22

`format` 选项用于控制章节标题的全局格式, 作用域为章节名字和随后的标题内容。可以用于控制章节标题的对齐方式、整体字体字号等格式。带加号的 `format+` 选项用于在已有的格式命令后附加内容。

`format` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字、编号和标题内容 (以及由 `nameformat`, `numberformat`, `aftername`, `titleformat`, `aftertitle`, `indent` 及 `hang` 选项设定的, 应用于这些内容之上的格式), 以实现特殊效果。

例如, 设置章标题为无衬线字体左对齐、为节标题增加无衬线字体设置、为小节标题加框 (`\fbox` 命令本身需带一个参数):

## 例 18

```
\ctexset{
  chapter/format      = \sffamily\raggedright,
  section/format      += \sffamily,
  subsection/format   += \fbox,
}
```

`format` 选项的默认设置见表 9。

表 9 `format` 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	<code>\Large\bfseries\centering</code>	<code>\raggedright</code> *
part (beamer)	同右	<code>\centering</code>
part	<code>\huge\bfseries\centering</code>	<code>\centering</code>
chapter	<code>\huge\bfseries\centering</code>	<code>\raggedright</code>
section (beamer)	同右	<code>\centering</code>
section	<code>\Large\bfseries\centering</code>	<code>\Large\bfseries</code>
subsection (beamer)	同右	<code>\centering</code>
subsection	同右	<code>\large\bfseries</code>
subsubsection	同右	<code>\normalsize\bfseries</code>
paragraph	同右	<code>\normalsize\bfseries</code>
subparagraph	同右	<code>\normalsize\bfseries</code>

\* 为了与  $\text{\TeX}2_{\epsilon}$  的默认效果保持一致, 在 `scheme = plain` 时, `part` 和 `chapter` 的 `nameformat` 和 `titleformat` 并不一样, 因此没有使用 `format` 选项统一设置名字和标题的格式。

---

```
.../nameformat * nameformat = {<格式命令>}
.../nameformat+ * nameformat+= {<格式命令>}
```

---

`nameformat` 用于控制章节名字的格式, 作用域为章节名字, 包括编号。它一般用于章节名 (包括编号) 与章节标题的字体、字号等设置不一致的情形。参见下面的 `titleformat` 选项。`nameformat+` 用于在已有的章节名字格式命令后附加内容。

`nameformat` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字和编号, 以实现特殊效果 (见例 24)。

`nameformat` 选项的默认设置见表 10。

表 10 nameformat 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\Large\bfseries
part (beamer)	同右	\usebeamerfont{part name} \usebeamercolor[fg]{part name}
part	{}	\huge\bfseries
chapter	{}	\huge\bfseries
section (beamer)	同右	\usebeamerfont{section name} \usebeamercolor[fg]{section name}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection name} \usebeamercolor[fg]{subsection name}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

---

```
.../numberformat * numberformat = {<格式命令>}
.../numberformat+ * numberformat+= {<格式命令>}
```

---

numberformat 选项用于控制章节编号的格式, 作用域仅为编号数字本身。对各级标题默认均为空, 当需要编号的格式和前后的章节名字不一样时可以使用。numberformat+ 用于在已有的编号格式命令后附加内容。

numberformat 选项的最后一个格式命令可以带有一个参数。这一参数用于接受编号数字。例如, 我们可以使用 numberformat 特别强调章标题中的数字:

例 19

```
\ctexset{
  chapter/number = \arabic{chapter},
  chapter/numberformat = \color{blue}\zihao{0}\emph,
}
```

上面的代码在 scheme = chinese 时可以做出类似这样的章标题效果:

## 第 4 章

numberformat 选项默认均设置为空, 故章节编号默认与章节名字使用相同的格式。

---

```
.../titleformat * titleformat = {<格式命令>}
.../titleformat+ * titleformat+= {<格式命令>}
```

---

`titleformat` 选项用于控制标题内容的格式, 作用域为章节标题内容。`titleformat+` 选项用于在已有的标题格式命令后附加内容。

`titleformat` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受标题内容。例如, 实现多行标题的居中悬挂对齐:

例 20

```
\usepackage{varwidth} %% 提供 varwidth 环境
\ctexset{
  chapter/name = {第,回},
  chapter/titleformat = \chaptertitleformat
}
\newcommand\chaptertitleformat[1]{%% 以标题内容为参数
  \begin{varwidth}[t]{.7\linewidth}#1\end{varwidth}}
.....
\chapter{情中情因情感妹妹\错里错以错劝哥哥}
```

上面的代码可以做出类似这样的章标题效果:

## 第三十四回 情中情因情感妹妹 错里错以错劝哥哥

`titleformat` 选项的默认设置见表 11。

表 11 `titleformat` 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\huge\bfseries
part (beamer)	同右	\usebeamerfont{part title}
part	{}	\Huge\bfseries
chapter	{}	\Huge\bfseries
section (beamer)	同右	\usebeamerfont{section title}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection title}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

---

```
.../aftername * aftername = {<代码>}
.../aftername+ * aftername+= {<代码>}
```

---

`aftername` 选项的参数<代码> 将被插入到章节编号与其后的标题内容之间, 用于控制格式变换。常用于控制章节编号与标题内容之间的距离, 或者控制标题是否另起一行。`aftername+` 用于在已有的代码后附加内容。该选项的默认设置见表 12。

---

```
.../aftertitle * aftertitle = {<代码>}
.../aftertitle+ * aftertitle+= {<代码>}
```

---

`aftertitle` 选项的参数<代码> 将被插入到章节标题内容之后。`aftertitle+` 用于在已有的代码后附加内容。该选项的默认设置见表 13。需注意, `sub3section` 或 `sub4section` 宏包选项(见 5.2 节)会影响 `aftertitle` 选项的默认值。

表 12 aftername 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\quad	\par\nobreak
part (beamer)	同右	\vskip 1em \par
part	同右	\par\vskip 20pt
chapter	\quad	\par\nobreak\vskip 20pt
section (beamer)	同右	\vskip 1em \par
section	同右	\quad
subsection (beamer)	同右	\vskip 1em \par
subsection	同右	\quad
subsubsection	同右	\quad
paragraph	同右	\quad
subparagraph	同右	\quad

表 13 aftertitle 选项的默认设置

标题名	默认值
part	\par
chapter	\par
section	\@@par
subsection	\@@par
subsubsection	\@@par
paragraph	{}
(sub3section)	\@@par
(sub4section)	同上
subparagraph	{}
(sub4section)	\@@par

part/pagestyle \* pagestyle = {<页面格式>}  
chapter/pagestyle \*

设置 book/ctexbook 或 report/ctexrep 文档类中, \part 与 \chapter 标题所在页的页面格式 (page style)。该选项的默认设置见表 14。

表 14 pagestyle 选项的默认设置

标题名	默认值
part (article)	无效
part	plain
chapter	plain

### 7.3 间距、缩进相关

section/runin \* runin = true|false  
subsection/runin \*  
subsubsection/runin \* runin 选项只对 \section 级以下标题有意义,用于确定标题与随后的正文是否排在同一段之  
paragraph/runin \* 上。该选项的默认设置见表 15。  
subparagraph/runin \*

默认情况下, \paragraph、\subparagraph 两级标题是与后面正文排在同一段的, runin 选项为 true; 但使用 sub3section 或 sub4section 宏包选项 (见 5.2 节) 后, 将对这两级标题设 runin 选项为 false, 这两级标题会改为排在不同段。

**注意:** section、subsection、subsubsection 的 aftertitle 默认值为 \@@par (分段命令), 这会导致 runin 选项失效。若需要对这些层级使用 runin = true, 必须同时将 aftertitle 设为空值:

```
\ctexset{section={runin=true, aftertitle={}}}
```

而 \paragraph、\subparagraph 的 aftertitle 默认值为 {} (空), 不存在此问题。

表 15 runin 选项的默认设置

标题名	默认值
part	无效
chapter	无效
section	false
subsection	false
subsubsection	false
paragraph	true
(sub3section)	false
(sub4section)	同上
subparagraph	true
(sub4section)	false

---

.../hang   ★ hang = true|false

Updated: 2020-04-23

hang 选项用于设置是否对章节标题实施悬挂缩进（缩进的宽度为名字宽度和 indent 选项设置的宽度之和）。

注意，当 hang = true 时，不恰当地设置选项 aftername 的值，可能会引发错误。这是因为当 hang = true 时， $\LaTeX$  内部会构造一个 `\hbox` 而进入受限水平模式（restricted horizontal mode）。若在 aftername 中加入包含 `\vskip` 等会导致从受限水平模式切出的垂直命令（vertical command）时，就会报错。特别地，aftername 的默认值也可能导致这种情形（见表 12）。因此，当设置 hang = true 时，用户必须恰当地设置选项 aftername 的值。

本选项对 beamer/ctexbeamer 文档类无效。对于 `\section` 级以下标题，若设置了 runin 选项为 true，即标题与随后正文排在同一段，hang 选项没有意义。该选项的默认设置见表 16。

表 16 hang 选项的默认设置

标题名	默认值
part	false
chapter	false
section	true
subsection	true
subsubsection	true
paragraph	无意义
(sub3section)	true
(sub4section)	true
subparagraph	无意义
(sub4section)	true

表 17 indent 选项的默认设置

标题名	默认值
part	0pt
chapter	0pt
section	0pt
subsection	0pt
subsubsection	0pt
paragraph	0pt
subparagraph	<code>\parindent</code>
(sub3section)	0pt
(sub4section)	同上

---

`.../indent` \* `indent = {<缩进间距>}`

---

Updated: 2020-04-23

`indent` 选项用于设置章节标题本身的首行缩进。该选项的默认设置见表 17。

如果 `indent` 的值是以 `em`、`ex` 或 `\ccwd` 为单位, 那么缩进间距的大小是相对于 `format` 中指定的字号大小。例如, 设置 `\part` 标题缩进三个字、`\section` 标题缩进 20pt:

例 21

```
\ctexset{
  part = {
    format += \raggedright,
    indent = 3\ccwd,
  },
  section = {
    format = \Large\bfseries,
    indent = 20pt,
  }
}
\part{首行缩进的标题}
\noindent 无缩进的正文。
\section{首行缩进的标题}
\noindent 无缩进的正文。
```

---

`.../beforeskip` \* `beforeskip = {<弹性间距>}`

---

`beforeskip` 选项用于设置章节标题前的垂直间距。该选项的默认设置见表 18。

**注意:** 若加载了 `titlesec` 宏包 (且未使用其 `loadonly` 选项), `titlesec` 将完全接管章节标题命令。此时 `beforeskip` 选项不再生效, 应改用 `titlesec` 的 `\titlespacing` 命令来设置标题前间距。

---

`.../afterskip` \* `afterskip = {<弹性间距>}`

---

`afterskip` 选项控制章节标题下方或后面的间距。

对于 `\section` 级以下标题, `runin` 选项会影响 `afterskip` 选项的意义: 若 `runin` 为 `true`, 标题与随后正文排在同一段, `<弹性间距>` 给出水平间距。否则, 正文另起一段, `<弹性间距>` 给出的是垂直间距。

该选项的默认设置见表 19。注意 `sub3section` 或 `sub4section` 宏包选项 (见 5.2 节) 会影响 `aftertitle` 选项的默认值。

**注意:** 若加载了 `titlesec` 宏包 (且未使用其 `loadonly` 选项), `titlesec` 将完全接管章节标题命令。此时 `afterskip` 选项不再生效, 应改用 `titlesec` 的 `\titlespacing` 命令来设置标题后间距。

---

`.../fixskip` \* `fixskip = true|false`

---

默认情况下, `article`、`book` 和 `report` 类的标题与正文的距离除了由 `beforeskip` 和 `afterskip` 选项设置的垂直间距外, 还会有一些多余的间距。`fixskip` 选项用于抑制这些多余间距。该选项默认不开启。

表 18 before skip 选项的默认设置

标题名	默认值
part (article)	4ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	50pt
section (beamer)	0pt
section	3.5ex plus 1ex minus .2ex
subsection (beamer)	0pt
subsection	3.25ex plus 1ex minus .2ex
subsubsection	3.25ex plus 1ex minus .2ex
paragraph	3.25ex plus 1ex minus .2ex
subparagraph	3.25ex plus 1ex minus .2ex

表 19 afterskip 选项的默认设置

标题名	默认值
part (article)	3ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	40pt
section (beamer)	0pt
section	2.3ex plus .2ex
subsection (beamer)	0pt
subsection	1.5ex plus .2ex
subsubsection	1.5ex plus .2ex
paragraph	1em
(sub3section)	1ex plus .2ex
(sub4section)	同上
subparagraph	1em
(sub4section)	1ex plus .2ex

```
.../break * break = {(格式命令)}
.../break+ * break+= {(格式命令)}
```

break 选项用于控制章节标题与之前正文的分隔关系。一般用于设置是否在标题之前分页或者设置行间罚点。break+ 用于在已有的格式命令后附加内容。

例如, 若当前页剩余高度小于正文高度的一半时, 则另起一页输出 \section 标题:

例 22

```
\usepackage{needspace}
\ctexset{section/break = \Needspace{.5\textheight}}
```

该选项的默认设置见表 20。

表 20 break 选项的默认设置

标题名	默认值
part (article)	{}
part	\if@openright\cleardoublepage\else\clearpage\fi
chapter	同上
section	\addpenalty{\@secpenalty}
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

```
.../afterindent * afterindent = true|false
```

afterindent 选项用于设置章节标题后首段的缩进。

book 和 report 类的 \part 标题被单独排在一页之上, afterindent 选项没有意义。对于 \section 级以下标题, 若设置了 runin 选项为 true, 即标题与随后正文排在同一段, afterindent 选项也就没有了意义。

该选项的默认设置见表 21。

表 21 afterindent 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	true	false
part	无效	无效
chapter	true	false
section	true	false
subsection	true	false
subsubsection	true	false
paragraph	true	false
subparagraph	true	false

## 7.4 目录、附录相关

---

`tocdepth` \* `tocdepth = <整数或章节名称>`

New: 2020-05-06

设置对章节标题编入目录的层次数。`tocdepth` 的值可以是一个整数,也可以是 `part`, `chapter` 等名称。层次数与名称的对应关系见表 5。章节层次的默认设置见表 6。

---

`.../tocline` \* `tocline = <{格式定义}>`

`tocline` 选项用于定义章节标题在目录文件 (.toc) 中的格式。`<格式定义>` 有两个参数: 参数#1 是 `part`、`chapter` 等名字, 参数#2 是标题内容。该选项的默认设置见表 22。

表 22 tocline 选项的默认设置

标题名	默认值
part	<code>\CTEXifname{\CTEXthepart\hspace{1em}}{ }#2</code>
chapter (chinese)	<code>\CTEXifname{\protect\numberline{\CTEXthechapter\hspace{.3em}}{ }#2</code>
chapter (plain)	<code>\CTEXnumberline{#1}#2</code>
section	<code>\CTEXnumberline{#1}#2</code>
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

这里 `\CTEXnumberline` 的意义是, 若标题#1 没有名字, 则不输出 `\numberline{\CTEXthe#1}` 等编号:

```
\CTEXifname{\protect\numberline{\cname CTEXthe#1\endcsname}}{ }
```

其中, `\CTEXifname` 的定义见 7.5 小节。

---

`chapter/lofskip` \* `lofskip = <{弹性间距}>`

`chapter/lotskip` \* `lotskip = <{弹性间距}>`

`lofskip` 选项控制插图目录 (.lof) 中, 章之间的插图标题的距离。同样, `lotskip` 选项控制表格目录 (.lot) 中, 章之间的表格标题的距离。

目前, 这两个选项只在 `chapter` 标题下有定义。它们的默认值, 在 `scheme` 选项的不同取值下都为 10 pt。

---

`appendix/numbering` \* `numbering = true|false`

控制是否对附录章(对应 `book` 与 `report`)或附录节(对应 `article`)进行编号, 用法与普通章节对应的 `numbering` 选项相同。该选项默认值为 `true`。

---

```
appendix/name * name = {<前名字>,<后名字>}
name = {<前名字>}
```

---

设置附录章(对应 `book` 与 `report`)或附录节(对应 `article`)的名字,用法与普通章节对应的 `name` 选项相同。

注意该选项与 `appendixname` 选项 (6.2 节) 在意义上有些重叠, 但不完全相同。`appendixname` 选项只用来重定义 `\appendixname`, 而不管 `\appendixname` 如何使用; 该选项则决定在章节标题中输出的名字, 可以调用 `\appendixname` 设置。

该选项的默认设置见表 23。

表 23 `appendix/name` 选项的默认设置

文档类	影响命令	scheme = chinese	实际定义	scheme = plain	实际定义
article	<code>\section</code>	{}		{}	
book, report	<code>\chapter</code>	<code>\appendixname\space</code>	附录 <sub>□</sub>	<code>\appendixname\space</code>	Appendix <sub>□</sub>

---

```
appendix/number * number = {<数字输出命令>}
```

---

设置附录章(对 `book` 与 `report`)或附录节(对 `article`)编号的数字输出格式,用法与普通章节对应的 `number` 选项相同。

`appendix/number` 选项同时也会控制附录章节计数器的交叉引用。与普通章节的 `number` 选项类似, 同样需要注意, 该选项不会影响计数器本身的输出, 即不影响 `\thesection` 或 `\thechapter` 的定义。

该选项的默认设置见表 24。

表 24 `appendix/number` 选项的默认设置

文档类	影响命令	默认值
article	<code>\section</code>	<code>\Alph{section}</code>
book, report	<code>\chapter</code>	<code>\Alph{chapter}</code>

## 7.5 辅助命令

$\text{\TeX}$  宏集还提供了一些辅助命令(宏), 用于存储章节标题格式, 或进行一些条件判断。

---

```
\CTEXthepart
\CTEXthechapter
\CTEXthesection
\CTEXthesubsection
\CTEXthesubsubsection
\CTEXtheparagraph
\CTEXthesubparagraph
```

---

以 $\text{\CTEXthe}$ 开头的这组宏给出结合了 `name` 与 `number` 选项的章节编号输出格式。例如在 `scheme = chinese` 时, 默认章编号输出格式就是 `\CTEXthechapter`, 形如“第一章”。

这组宏在  $\text{\TeX}$  文档类中将代替 `\thechapter` 等宏的作用, 在章节中引用本章节的完整编号。例如用于帮助定义自定义的目录格式、页眉格式等。

---

`\CTEXifname` `\CTEXifname {<有名字时的内容>} {<无名字时的内容>}`

`\CTEXifname` 会根据当前章节有无名字展开得到不同内容 (通常是格式命令)。由于章节名字总是与编号一起出现, 章节有无名字通常也表达为“章节是否编号”。在  $\text{\LaTeX}$  中, 后者取决于以下几个方面: 章节深度是否不大于计数器 `secnumdepth` 的值, 章节标题是否使用不带星号的命令。在  $\text{\CTEX}$  宏集中, 后者还取决于 `.../numbering` 是否为 `true`。

`\CTEXifname` 可用于 `format`, `titleformat`, `aftertitle`, `afterskip`, `indent` 这五个选项和 `\chapter` 标题 `beforeskip` 选项的格式设置之中。也可用于帮助定义自定义的目录格式、页眉格式等。

例如, 设置章的标题有名字时左对齐, 无名字时居中对齐, 并且在标题后画一条横线。

例 23

```
\ctexset{
  chapter/format      = \CTEXifname{\raggedright}{\centering},
  chapter/aftertitle = \par\CTEXifname{}{\hrule},
}
```

## 7.6 示例

我们最后举一个稍微复杂的例子, 来看看上述选项的综合应用。

例 24

```
\ctexset {
  chapter = {
    beforeskip = 0pt,
    fixskip    = true,
    format     = \Huge\bfseries,
    nameformat = \rule{\linewidth}{1bp}\par\bigskip\hfill\chapternamebox,
    number     = \arabic{chapter},
    aftername  = \par\medskip,
    aftertitle = \par\bigskip\nointerlineskip\rule{\linewidth}{2bp}\par
  }
}
\newcommand\chapternamebox[1]{%
  \parbox{\ccwd}{\linespread{1}\selectfont\centering #1}}
.....
\chapter{熟悉 \LaTeX}
```

本例的设置效果大致如下:

第  
1  
章

熟悉  $\text{\LaTeX}$

---

## 第 8 节 实用命令

### 8.1 字号与间距

---

`\zihao` `\zihao {<字号>}`

用于调整字号大小。其中 `<字号>` 的有效值共有 16 个, 如表 25 所示。使用 `\zihao` 命令调整字体大小时, 西文字号大小会始终和中文字号保持一致。

表 25 中文字号

〈字号〉	大小 (bp)	大小 (pt)	意义
0	42	42.157 49	初号
-0	36	36.135	小初号
1	26	26.097 49	一号
-1	24	24.09	小一号
2	22	22.082 49	二号
-2	18	18.067 49	小二号
3	16	16.06	三号
-3	15	15.056 24	小三号
4	14	14.052 49	四号
-4	12	12.045	小四号
5	10.5	10.539 37	五号
-5	9	9.033 74	小五号
6	7.5	7.528 12	六号
-6	6.5	6.524 37	小六号
7	5.5	5.520 61	七号
8	5	5.018 74	八号

表 25 中的字号大小基于默认的 word 字号系统。当启用 `experiment/font-size-system = letterpress` 或用户自定义字号系统时, 部分字号的 bp 数值会发生变化, 详见 5.1 节。

---

`\ziju` `\ziju` `{(中文字符宽度的倍数)}`

用于调整相邻汉字之间的间距, 即(在正常中文行文中)前一个汉字的右边缘与后一个汉字的左边缘之间的距离。其中参数可以是任意浮点数值; 而中文字符宽度指的是实际汉字的宽度, 不包含当前字距。

这个命令会影响 `\ccwd` 的值, 但不会影响英文字距。

---

`\ccwd` 当前汉字的字宽保存在长度寄存器 `\ccwd` 之中。汉字字宽是相邻两个汉字中心之间的距离, 包含字距在内。因此修改字距会间接修改字宽。

## 8.2 中文数字转换

CT<sub>E</sub>X 宏集的中文数字转换功能实际上是调用 `zhnumber` 宏包来完成。下面只介绍一些基本的用法, 更高级的用法可以查阅 `zhnumber` 宏包的文档。

---

`\chinese` `\chinese` `{(counter)}`  
`\pagenumbering` `{chinese}`

`\chinese` 命令与 `\roman` 等命令的用法类似, 作用在一个 L<sup>A</sup>T<sub>E</sub>X 计数器上, 将计数器的值以中文数字的形式输出。

---

`\zhnumber` `\zhnumber` `{(number)}`

以中文格式输出数字。这里的数字可以是整数、小数和分数。

---

`\zhdigits` `\zhdigits` `{(number)}`

将阿拉伯数字转换为中文数字串。

---

`\CTEXnumber` `\CTEXnumber` `\(macro)` `{(number)}`

`\(macro)` 必须是一个 T<sub>E</sub>X 宏, 不需预先定义。`\CTEXnumber` 通过 `\zhnumber` 将 `(number)` 转为中文数字, 最后将结果存储在 `\(macro)` 里。对 `\(macro)` 的定义是局部的, 将它展开一次就可以得到转换结果。

一般来说, 并不需要使用 `\CTEXnumber`, 直接使用 `\zhnumber` 即可。但是, 如果在文档中需要多次使用同一个数字 `(number)` 的中文形式, 就可以先用 `\CTEXnumber` 将结果保存起来备用, 而不是每次使用时都用 `\zhnumber` 现场转换一次。

---

`\CTEXdigits` `\CTEXdigits` `\(macro)` `{(number)}`

`\CTEXdigits` 与 `\CTEXnumber` 类似, 但其转换的结果是中文数字串, 而不是中文数字。

## 8.3 杂项

---

`\CTeX` 用于显示 CT<sub>E</sub>X 标志。

## 第 9 节 *Lua $\LaTeX$* 下的中文支持方式

在 *Lua $\LaTeX$*  下,  $\text{CT}_{\text{E}}\text{X}$  宏集依赖 *LuaTeX-j*a 宏包来完成中文支持。该宏包是日本  $\text{T}_{\text{E}}\text{X}$  社区的北川弘典、前田一贵、八登崇之等人开发的, 设计目的主要是在 *Lua $\text{T}_{\text{E}}\text{X}$*  引擎下实现日本  $\text{p}_{\text{T}}\text{E}\text{X}$  引擎的(大部分)功能。它为了兼容  $\text{p}_{\text{T}}\text{E}\text{X}$  的使用习惯, 对  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  的  $\text{NFSS}$  作了不少修改和扩充。这对于简体中文用户来说不是必要的, 因而  $\text{C}_{\text{T}}\text{E}\text{X}$  禁用了它在  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  格式下的大部分设置, 只保留了必要的部分。同时修改了它的字体设置方式, 使得相关命令与 *xeCJK* 宏包大致相同。

20150420 版以后的 *LuaTeX-j*a 宏包开始支持竖排, 但  $\text{C}_{\text{T}}\text{E}\text{X}$  暂不支持竖排。

### 9.1 *Lua $\LaTeX$* 下替代字体的设置

---

```
AlternateFont \setCJKfamilyfont {<family>} {<base font name>}
[
  <base font features> ,
  AlternateFont =
  {
    {<character range1>} {<alternate font name1>} ,
    {<character range2>} {<alternate font name2>} {<alternate font features2>} ,
    .....
  }
]
```

---

Updated: 2020-04-30

在设置字体族  $\langle family \rangle$  的时候, 同时设置该字体族在字符范围  $\langle character\ range_n \rangle$  内, 对应字形的替代字体。

---

```
CharRange \setCJKfamilyfont {<family>} {<alternate font name>}
[
  CharRange = {<character range>} ,
  <alternate font features>
]
```

---

只设置字体族  $\langle family \rangle$  在字符范围  $\langle character\ range \rangle$  内, 对应字形的替代字体。

一个  $\backslash\text{setCJKfamilyfont}$  里只能使用一次 *CharRange* 或者 *AlternateFont*, 但可以将它们分开重叠使用。例如下面的方式是有效的。

#### 例 25

```
\setCJKmainfont [AlternateFont={...}{...}, ...]{...}
\setCJKmainfont [CharRange={"4E00->"67FF,-2}, ...]{...}
\setCJKmainfont [CharRange={"6800->"9FFF}, ...]{...}
```

---

```
declarecharrange * \ctexset
{
  declarecharrange =
  {
    {<name1>} {<character range1>} ,
    {<name2>} {<character range2>} ,
    ...
  }
}
```

---

New: 2020-04-30

预先声明字符范围。声明字符范围  $\langle name \rangle$  之后, 它的名字  $\langle name \rangle$  可以用在 *AlternateFont* 和 *CharRange* 选项的  $\langle character\ range \rangle$  之中, 表示对应的字符范围。

在声明字符范围  $\langle name \rangle$  的同时, 还为  $\backslash\text{setCJKmainfont}$  等字体设置命令定义了选项  $\langle name \rangle$ , 用于设置对应字符的替代字体:

```
<name> = {<alternate font name>}
<name> = {<alternate font name>} {<alternate font features>}
```

$\langle name \rangle$  选项可以与 `AlternateFont` 共同使用,但不能与 `CharRange` 一起使用。如果没有给  $\langle name \rangle$  设置值,则等价于设置 `CharRange= $\langle name \rangle$` ,即只设置  $\langle name \rangle$  对应的字符范围的替代字体。

---

```
clearalternatefont * \ctexset
resetalternatefont * {
    clearalternatefont = {\family1, family2, ...} ,
    resetalternatefont = {\family1, family2, ...} ,
    clearalternatefont ,
    resetalternatefont
}
```

---

清除与重置 CJK 字体族  $\langle family \rangle$  的替换字体设置。如果没有给定值,则作用于当前 CJK 字体族。清除与重置操作总是全局的。

## 第 10 节 C<sub>T</sub>E<sub>X</sub> 宏集的配置文件

C<sub>T</sub>E<sub>X</sub> 宏集提供了不同的配置文件,可以通过修改配置文件来改变 C<sub>T</sub>E<sub>X</sub> 宏集默认行为。

在多数情况下,并不需要修改配置文件,C<sub>T</sub>E<sub>X</sub> 宏集的默认设置已经能满足大多数用户的需要。不恰当地修改 C<sub>T</sub>E<sub>X</sub> 宏集的默认行为也可能导致同一文件在别处无法正常编译或排版效果完全不同,因此修改应该慎重。

但在一些情况下,直接修改配置文件仍是必要的,例如:

- 系统没有安装默认设置的字体文件,无法编译。
- 需要经常编译来自其他系统的中文 T<sub>E</sub>X 文件,但对方的操作系统或默认设置与本机不同。

与 C<sub>T</sub>E<sub>X</sub> 宏集的源代码一样,配置文件采用 L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> 的语法编写。

C<sub>T</sub>E<sub>X</sub> 宏集的配置文件随宏包其他文件一起安装在 T<sub>E</sub>X 系统 TDS 目录树中,文件后缀是 `.cfg`。为了避免本地配置文件内容因 C<sub>T</sub>E<sub>X</sub> 宏集的更新而丢失,不要直接修改系统 TDS 目录树中的配置文件,而应该将系统自带的配置文件复制到本地的或用户私有的 TDS 目录树中修改,并运行 `texhash` 命令刷新文件名数据库。

例如对于 T<sub>E</sub>X Live,系统自带的配置文件就在 T<sub>E</sub>X Live 安装目录下的 `texmf-dist/tex/latex/ctex/config/` 子目录下,可以修改它的副本,保存在本地 TDS 树的 `texmf-local/tex/latex/ctex/` 目录下,或者用户 TDS 树的 `~/texmf/tex/latex/ctex/` 目录下,作为本地/用户专有的配置文件。复制配置文件后需要运行 `texhash` 命令使本地配置文件生效。

MiK<sub>T</sub>E<sub>X</sub> 的配置文件也保存在类似的目录结构中,MiK<sub>T</sub>E<sub>X</sub> 管理的几个 TDS 根目录可以在 MiK<sub>T</sub>E<sub>X</sub> Options 设置项中查看到,这里不再赘述。

除了修改本地 T<sub>E</sub>X 系统中的配置文件,对于特定文档,也可以将修改过的配置文件保存在文档的工作目录下。此时配置文件就只对工作目录下的所有文档生效。

### 10.1 修改宏包默认选项

配置文件 `ctexopts.cfg` 可以用来修改宏包的默认选项。随系统安装的配置文件除了文件信息声明外没有实际的内容,但在注释中给出了一个简单的示例,只要取消注释就可以生效。

#### 例 26

```
% 系统自带 ctexopts.cfg 注释中的示例语句,固定默认字体集为 windows。
% 该设置可以用在安装了 Windows 字体的非 Windows 系统中。
\ctex_set:nn { option } { fontset = windows }
```

如上例所示,宏包选项通常使用 L<sup>A</sup>T<sub>E</sub>X3 的 `\ctex_set:nn` 命令完成键值设置,第一个参数是固定的子模块 option,第二个参数中是用户定义的新的默认宏包选项。

`ctexopts.cfg` 中的设置将在 C<sub>T</sub>E<sub>X</sub> 宏集的开始处,定义过宏包选项之后,`\ProcessKeysOptions` 命令之前生效。最好只使用此配置文件修改宏包默认选项。

## 10.2 宏包载入后的配置

配置文件 `ctex.cfg` 将在宏包的末尾被载入生效。可以用它完成任意的设置,或是覆盖已有的定义。随系统安装的配置文件除版本信息外没有实际内容,注意配置文件中也使用 L<sup>A</sup>T<sub>E</sub>X3 语法。

### 例 27

```
% 简单的 ctex.cfg 内容示例。
% 修改默认的页面格式设置。
\pagestyle{plain}
```

### 例 28

```
% 略复杂的 ctex.cfg 内容示例：禁止段末孤字成行。
% 在使用 XeTeX 编译时，打开 xeCJK 的 CheckSingle 选项。
\sys_if_engine_xetex:T
{
  \xeCJKsetup { CheckSingle }
}
% 在使用 LuaTeX 编译时，设置 LuaTeX-ja 的 jcharwidowpenalty 参数。
\sys_if_engine_luatex:T
{
  \ltjsetparameter { jcharwidowpenalty = 10000 }
}
```

## 10.3 配置标题中文翻译

由于 C<sub>T</sub>E<sub>X</sub> 宏集需要同时支持 GBK 和 UTF-8 两种编码,因此对标题的中文翻译写在两个配置文件当中:`ctex-name-gbk.cfg` 和 `ctex-name-utf8.cfg`。两个文件的设置相同,只是编码不同。

为了同一文档在不同电脑上编译效果的一致性,通常不建议修改默认的中文翻译。

## 10.4 自定义字体集

4.3 节介绍的用于 `fontset` 选项的自定义字库文件,类似于 C<sub>T</sub>E<sub>X</sub> 宏集的配置文​​件,也应该与其他本地配置文件一起保存在本地 TDS 目录树下,并可以配合 `ctexopts.cfg` 等配置文件使用。

## 第 11 节 第三方宏包兼容

C<sub>T</sub>E<sub>X</sub> 宏集为使 `\labelformat` 功能与章节标题设置兼容,对 `cleveref` 等宏包进行了补丁。由于 `cleveref` 长期未维护(CTAN 上仍为 0.21.4),与新版 L<sup>A</sup>T<sub>E</sub>X 内核和 `hyperref` 存在兼容问题,C<sub>T</sub>E<sub>X</sub> 提供了选项以控制补丁行为。

---

```
patch/cleveref * \ctexset | patch/cleveref = true || false |
```

---

New: 2026-04-24

控制是否对 `cleveref` 宏包应用兼容补丁。默认为 `true`。该补丁在 `cleveref` 的 `\refstepcounter` 变体中注入 `\expandafter`，以确保 `\labelformat` 格式化后的标签能正确展开。

当该补丁与上游  $\LaTeX$  内核或 `hyperref` 的更新产生冲突时，可以将此选项设为 `false` 以关闭补丁。

**注意：**当同时使用 `hyperref` 和 `cleveref` 时，`\appendix` 之后的交叉引用类型可能不正确（例如，`appendix` 退化为 `chapter`）。这是上游  $\LaTeX$  内核 `firstaid` 对 `cleveref` 补丁不完整所致，与  $\CTeX$  无关。可以使用以下方法规避：

## 例 29

```
\AddToHook{cmd/appendix/before}{%
  \crefalias{chapter}{appendix}%
  \crefalias{section}{subappendix}%
  \crefalias{subsection}{subsubappendix}%
  \crefalias{subsubsection}{subsubsubappendix}%
}
```

## 第 12 节 对旧版本的兼容性

### 12.1 $\CTeX$ 0.8a 及以前的版本

在 `ctex-kit` 项目成立之前， $\CTeX$  宏包的最后一个版本是  $\CTeX$  0.8a (2007/05/06)。

第 2 版未考虑对这些很早版本的兼容性。

### 12.2 $\CTeX$ 0.9– $\CTeX$ 1.0d

在 2009 年在 `ctex-kit` 项目成立后，新增了  $X_{\LaTeX}$  引擎的支持，并增加了不少控制字体的命令和选项。

这里主要介绍新版本  $\CTeX$  宏包相对 1.02d 版本 (2014/06/09) 的兼容性。

第 2 版的  $\CTeX$  宏包已尽力保证对 1.0x 版本的兼容性，原有为 1.0x 编写的代码，在第 2 版的  $\CTeX$  宏包下保证仍能编译，并且在大多数情况下保持编译效果不变。

$\CTeX$  宏包在 0.8a 以前的版本支持以 `CCT` 作为底层中文支持方式，从 0.9 版之后即不再推荐使用，只保留向后兼容。在  $\CTeX$  宏包第 2 版中则完全不再支持 `CCT`。

下面这些是在旧版本  $\CTeX$  宏包中存在，而在新版本中已不建议使用的选项和命令，在未来版本中可能会删去它们的支持。

在多数情况下它们的功能仍将保留，但也有部分选项命令功能已失效。

---

`cs4size`    分别相当于 `zihao=-4` 和 `zihao=5`，过时选项。  
`c5size`

---



---

`CCT`        相关选项已删除。  
`CCTfont`

---



---

`indent`    `indent` 和 `noindent` 什么也不做，过时选项。  
`noindent`

---

在中文版式下，`ctex` 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开。 $\CTeX$  文档类的相关功能由章节标题的 `afterindent` 选项的值来确定。

zhmap    nozhmap    zhmap 宏包选项增加了参数, 扩充了功能, 除了支持真假值参数外, 还支持选择 zhmCJK 作为底层中文处理宏包。(4.3 节)

nozhmap 选项相当于zhmap=false。过时选项。

winfonts    adobefonts    nofonts    宏包选项 winfonts 相当于fontset=windows, adobefonts 相当于fontset=adobe, nofonts 相当于fontset=none。这几个选项是过时选项, 对于新文档, 应使用 fontset 选项设置不同字体集。

另外, 第 2 版 C<sub>T</sub>E<sub>X</sub> 宏包的默认字体不再是 Windows 系统字体, 而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(4.3 节)。

punct    nopunct    旧版本中宏包 punct 选项没有参数, 现在可以用参数设定标点风格(5.3 节)。原有无参形式的 punct 选项相当于punct=quanjiao。

旧版宏包中 nopunct 选项的效果大致相当于punct=plain。过时选项, 不推荐使用。

cap    nocap    原有的 cap 和 nocap 选项由新的 scheme 选项代替。(5.3 节)

cap 选项相当于scheme = chinese, nocap 选项相当于scheme = plain。它们均已过时, 仅因兼容性而保留。

space    nospace    新版本宏包 space 选项增加真假值参数。(5.3 节)

nospace 选项相当于space=false, 成为过时选项。

fancyhdr    新版本宏包中总是自动处理对 fancyhdr 宏包的兼容性, 而由用户自己使用 \usepackage 载入 fancyhdr 宏包。

fancyhdr 选项过时, 因兼容性保留, 功能是载入 fancyhdr 宏包。

hyperref    新版本宏包中总是自动处理对 hyperref 宏包的兼容性, 而由用户自己使用 \usepackage 载入 hyperref 宏包。

hyperref 选项过时, 因兼容性保留, 功能是在导言区末尾载入 hyperref 宏包。

fntef    旧版本的 fntef 选项用于统一 CCTfntef 与 CJKfntef 的界面, 新版本 C<sub>T</sub>E<sub>X</sub> 宏集不再支持 CCT, 也不再自动载入 CJKfntef 或 xeCJKfntef 宏包, 而仅在其末尾做适当格式调整。

fntef 选项过时, 因兼容性保留, 功能是根据引擎载入 CJKfntef (pdf<sub>T</sub>E<sub>X</sub>) 或 xeCJKfntef (X<sub>g</sub><sub>T</sub>E<sub>X</sub>) 宏包。

\CTEXunderdot    \CTEXunderline    \CTEXunderdblline    \CTEXunderwave    \CTEXsout    \CTEXxout    CTEXfilltwosides    在调用 fntef 宏包选项的同时, 旧版本 C<sub>T</sub>E<sub>X</sub> 宏包由于需要支持 CCT 系统, 会将以\CJK 开头的 \CJKunderline 等宏换名为以\CTEX 开头的 \CTEXunderline 等宏。此功能在新版本的 C<sub>T</sub>E<sub>X</sub> 宏集中已失去意义。此外, 在pdf<sub>T</sub>E<sub>X</sub> 引擎下, 用于设置格式的 \CJKunderdotbasesep 等宏也被更名为 \CTEXunderdotbasesep 等宏。

在新版本中, 上述由 fntef 衍生的相关命令和环境均被移除。

\CTEXsetfont    更新当前的中文字体信息, 包括当前字距(\ccwd)和段首缩进(\parindent)。一般来说, 用户无需使用这个命令。

\CTEXindent    更新 \ccwd 宽度后设置\parindent=2\ccwd。过时命令。

\CTEXnoindent    设置\parindent=0pt。过时命令。

---

`\CTEXsetup` `\CTEXsetup[⟨选项⟩]{⟨标题⟩}`

相当于设置了 `\ctexset{⟨标题⟩ = {⟨选项⟩}` }。过时命令。

---

`\CTEXoptions` `\CTEXoptions[⟨选项⟩]`

相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

---

`\Chinese` `\Chinese{⟨counter⟩}`

新版宏集中 `\chinese` 统一了旧版本中 `\chinese` 和 `\Chinese` 的功能。因此, 该命令已过时。

---

`captiondelimiter` 原为 `\CTEXoptions` 命令的选项, 用于控制 `\caption` 编号后面的标点。此选项已过时, 并在新版本的 `CTEX` 宏包中失效。

可以使用 `caption` 宏包的 `labelsep` 选项来完成同样的功能。

### 例 30

```
% 代替 \CTEXoptions[captiondelimiter={:}]
\usepackage{caption}
\captionsetup{labelsep=colon}
```

## 12.3 CT<sub>E</sub>X 1.02c 以后的 SVN 开发版

CT<sub>E</sub>X 宏包在 1.02c 版本(2011/03/11)之后在 Google code 上的 SVN 开发版本, 内部版本号一直升到 1.11 版, 但从未正式发布。SVN 开发版在 1.02c 版本的基础上新增的功能在第 2 版中大多继承了过来, 但新增的命令与选项都不再保持兼容。

CT<sub>E</sub>X 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

## 12.4 CT<sub>E</sub>X 2.2 之前的版本

---

part/beforeskip  
chapter/beforeskip  
section/beforeskip  
subsection/beforeskip  
subsubsection/beforeskip  
paragraph/beforeskip  
subparagraph/beforeskip

在 CT<sub>E</sub>X 2.2 之前的版本中, `beforeskip` 选项的符号还用于确定章节标题后首段的缩进。当 `beforeskip` 是负值时, 章节标题后的第一段按英文文档的排版习惯, 没有首行缩进, 否则保留首行缩进。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 `afterindent` 选项来设置。如果原先设置 `beforeskip` 为负值, 在新版本中需要改为正值, 并设置相应的 `afterindent` 选项为 `false`。

---

section/afterskip  
subsection/afterskip  
subsubsection/afterskip  
paragraph/afterskip  
subparagraph/afterskip

在 CT<sub>E</sub>X 2.2 之前的版本中, 对于 `\section` 级以下标题, `afterskip` 选项的符号用于确定标题与随后正文是否排在同一段。如果是正值, 则正文另起一段, 否则标题与随后正文排在同一段, `afterskip` 的绝对值给出水平间距。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 `runin` 选项来设置。如果原先设置 `afterskip` 为负值, 在新版本中需要改为正值, 并设置相应的 `runin` 选项为 `true`。

## 12.5 CT<sub>E</sub>X 2.4.1 和 2.4.2

---

part/fixbeforeskip  
chapter/fixbeforeskip

这两个选项已经被删除, 相应功能由新的选项 `fixskip` 提供。

## 12.6 C<sub>T</sub>E<sub>X</sub> 2.5 之前的版本

C<sub>T</sub>E<sub>X</sub> 2.5 有一些比较大的变动。

UTF8  
GBK (pdf)L<sup>A</sup>T<sub>E</sub>X 格式下, 文档编码初始值统一设置成 UTF-8。因此, 仍旧使用 GBK 编码的文档, 需要在文档类或宏包选项中显式指定 GBK。

\CTEXunderdot  
\CTEXunderline  
\CTEXunderdblline  
\CTEXunderwave  
\CTEXsout  
\CTEXxout  
CTEXfilltwosides 不再默认载入 CJKfntef 或 xeCJKfntef 宏包, 同步移除有关命令和环境。若需使用相关宏包, 建议用户使用 `\usepackage` 命令主动载入。

fntef 作为兼容性保留, 会视编译引擎载入相应宏包。

windows 不再支持 Windows XP 系统, 默认要求 Windows 系统有微软雅黑字体。建议 Windows XP 系统的用户及时更新操作系统。若一定要在 Windows XP 中使用, 请使用 2.5 以前的版本。

windowsold  
windowsnew 过时字库选项, 作为兼容性保留, 功能是载入 windows 字库。

ubuntu 改用思源(Noto CJK)和文鼎字库。该字库不再支持pdfL<sup>A</sup>T<sub>E</sub>X编译。

AlternateFont 不再支持将替代字体的可选项放在字体名之前的方括号中, 新的语法是将可选项放在字体名之后的花括号之内。

除了以上列出的选项以外, 当用户使用 C<sub>T</sub>E<sub>X</sub> 系列文档类, 且使用 L<sup>A</sup>T<sub>E</sub>X 或 upL<sup>A</sup>T<sub>E</sub>X 编译时, 若用户没有在文档类选项中显式指定 dvips/dvipdfmx/dvisvgm 等驱动选项, 则文档类指定默认驱动为 DVIPDFM<sub>x</sub>。

## 第 13 节 宏集依赖情况与手工安装方法

本节介绍 C<sub>T</sub>E<sub>X</sub> 宏集的依赖情况, 并介绍手工编译安装的具体方法。通常用户只需参照第 2.2 节介绍的方法, 使用发行版自带的包管理器安装本宏集。

C<sub>T</sub>E<sub>X</sub> 宏集有两个源文件: `ctex.dtx`、`ctexpunct.spa`。使用不同的编译方式时, C<sub>T</sub>E<sub>X</sub> 依赖的宏包略有不同。在手工安装 C<sub>T</sub>E<sub>X</sub> 宏集之前, 请确保你的 T<sub>E</sub>X 发行版中已经正确安装了这些宏包。C<sub>T</sub>E<sub>X</sub> 依赖宏包的详情叙述如下:

- `expl3`、`xparse` 和 `l3keys2e` 宏包。它们属于 `l3kernel` 和 `l3packages` 宏集。
- `indentfirst` 宏包, 属于 `tools` 宏集。
- `zhnumber` 宏包。
- ➔ 以上是各种编译方式都必需的依赖项。
- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包, 它还依赖
  - `iftex` 宏包。

- ltxcmds 宏包。
- kvoptions 宏包。
- kvsetkeys 宏包。
- keyval 宏包, graphics 宏集。
- ➔ 以上是使用pdfL<sup>A</sup>T<sub>E</sub>X 或 L<sup>A</sup>T<sub>E</sub>X + DVIPDFMx 的编译方式所需要的依赖项,其中 zhmCJK 是可选的。
- xeCJK 宏集, 它还依赖
  - xtemplate 宏包, 它属于 l3packages 宏集。
  - fontspec 宏包。
- ➔ 以上是使用 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 编译时的依赖项。
- luatexja 宏包, 它还依赖
  - adobemapping 宏包。
  - luaotfload 宏包, 它还依赖 lualibs 宏包。
  - luatexbase 宏包, 它还依赖 ctablestack 宏包。
  - atbegshi 宏包。
  - etoolbox 宏包。
  - iftex 宏包。
  - infwarerr 宏包。
  - ltxcmds 宏包。
  - pdftexcmds 宏包。
  - xkeyval 宏包。
  - chinese-jfm 宏包。
- fontspec 宏包。
- ➔ 以上是使用 LuaL<sup>A</sup>T<sub>E</sub>X 编译时的依赖项。
- zhmetrics-uptex 宏包。
- ➔ 以上是使用 upL<sup>A</sup>T<sub>E</sub>X 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 T<sub>E</sub>X Live 和 MiK<sub>T</sub>E<sub>X</sub>。因此, 若你希望使用 zhmCJK 作为 C<sub>T</sub>E<sub>X</sub> 宏集的底层中文支持方式, 那么你需要自行安装该宏包。zhmCJK 的安装较为复杂。我们建议你

1. 从 CTAN 下载 zhmCJK 宏包的 **TDS 安装包**,
2. 按目录结构将文件复制到 T<sub>E</sub>X 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 T<sub>E</sub>X 发行版的 ls-R 数据库以完成安装。

其他细节, 可参照其**宏包手册**中第 3 节的指导。

C<sub>T</sub>E<sub>X</sub> 宏集已被 T<sub>E</sub>X Live 和 MiK<sub>T</sub>E<sub>X</sub> 收录, 若无特别理由, 我们强烈建议用户使用包管理器安装本宏集。

若要手工安装, 请遵循如下步骤:

1. 从 CTAN 下载 C<sub>T</sub>E<sub>X</sub> 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 T<sub>E</sub>X 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 T<sub>E</sub>X 发行版的 ls-R 数据库以完成安装。

## 第 14 节 开发人员

- 吴凌云 (aloft@ctex.org)
- 江疆 (gzjjgod@gmail.com)

- 王越 (yuleopen@gmail.com)
- 刘海洋 (LeoLiu.PKU@gmail.com)
- 李延瑞 (LiYanrui.m2@gmail.com)
- 陈之初 (zhichu.chen@gmail.com)
- 李清 (sobenlee@gmail.com)
- 黄晨成 (liamhuang0205@gmail.com)
- 曾祥东 (xdzeng96@gmail.com)
- 李泽平 (zepinglee@gmail.com)
- 周宇恺 (muzimuzhi@gmail.com)
- 张瑞熹 (ruixizhang42@gmail.com)

## 参考文献

- [1] DONALD ERVIN KNUTH. *The T<sub>E</sub>Xbook, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBACH and MICHEL GOOSSENS. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

## 第 15 节 代码实现

```

1 <@@=ctex>
2 <*class|style>
3 \cs_if_exist:NF \NewDocumentCommand
4 { \RequirePackage { xparse } }
5 \prop_gput:Nnn \g_msg_module_type_prop { ctex } { Class }
6 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexart }
7 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexbook }
8 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexrep }
9 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexbeamer }
10 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexsize }
11 \prop_gput:Nnn \g_msg_module_name_prop { ctex } { ctexheading }
12 </class|style>
13 <*class|ctex>

```

检查 expl3 的版本。

```

14 \msg_new:nnnn { ctex } { l3-too-old }
15 { Support~package~`#1'~too~old. }
16 {
17   Please~update~an~up-to-date~version~of~the~bundles\\
18   `l3kernel'~and~`l3packages'\\
19   using~your~TeX~package~manager~or~from~CTAN.
20 }
21 \@ifpackagelater { expl3 } { 2025/10/09 } { }
22 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }

```

`\c__ctex_engine_str` 引擎检查。目前 L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> 将 A<sup>P</sup>T<sub>E</sub>X 识别为 u<sup>P</sup>T<sub>E</sub>X。

```

\c__ctex_engine_file_str
23 \str_const:Ne \c__ctex_engine_str
24 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }
25 \msg_new:nnnn { ctex } { engine-not-supported }
26 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
27 { You~can~switch~to~xelatex,~lualatex,~pdfplatex,~uplax,~or~aplatex. }
28 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }
29 {
30   \str_const:Ne \c__ctex_engine_file_str
31   { ctex-engine- \c__ctex_engine_str .def }
32 }
33 { \msg_critical:nne { ctex } { engine-not-supported } { \c__ctex_engine_str } }
34 </class|ctex>
35 <*class|ctex|ctexheading|ctexsize>

```

`ctexsize` 也要载入 `fix-cm` 包解决传统 cm 字体字号缺失的问题。

```

<!ctexsize> 36 \RequirePackage { ctexhook , ctexpatch }
<!ctexheading> 37 \RequirePackage { fix-cm }

```

宏包载入检查。

```

38 <*class|ctex>
39 <*class>
40 \ctex_disable_package:n { ctex }
41 \ctex_disable_package:n { ctexcap }
42 </class>
43 \ctex_disable_package:n { ctexsize }
44 \ctex_disable_package:n { ctexheading }
45 </class|ctex>

```

`\c__ctex_everysel_loaded_bool` L<sup>A</sup>T<sub>E</sub>X 2021-06-01 以后的版本内建了 `everysel` 包的功能。

```

46 <*!ctexsize>

```

```

47 <!*ctexheading>
48 \ctex_if_format_at_least:nTF { 2021/06/01 }
49   { \bool_const:Nn \c__ctex_everysel_loaded_bool { \c_false_bool } }
50   {
51     \RequirePackage { everysel }
52     \bool_const:Nn \c__ctex_everysel_loaded_bool { \c_true_bool }
53   }
54 </!ctexheading>

```

## 15.1 内部函数与变量

```

\l__ctex_tmp_tl 临时变量。
\l__ctex_tmp_int
\l__ctex_tmp_box 55 \tl_clear_new:N \l__ctex_tmp_tl
\l__ctex_tmp_dim 56 \int_new:N \l__ctex_tmp_int
                    57 \box_new:N \l__ctex_tmp_box
                    <!ctexheading> 58 \dim_new:N \l__ctex_tmp_dim

```

```

\ctex_define_option:n 在宏包内部使用的键值选项定义、设置命令。
\ctex_define:n
  \ctex_set:n 59 </!ctexsize>
  \ctex_set:nn 60 \cs_new_protected:Npn \ctex_define_option:n
                61   { \keys_define:nn { ctex / option } }
                62 <!*ctexsize>
                63 \cs_new_protected:Npn \ctex_define:n
                64   { \keys_define:nn { ctex } }
                65 \cs_new_protected:Npn \ctex_set:n
                66   { \keys_set:nn { ctex } }
                67 \cs_new_protected:Npn \ctex_set:nn #1
                68   { \keys_set:nn { ctex / #1 } }

```

\ctex\_scheme\_input:n 输入 scheme 文件。先查找当前文档类下的<scheme>,找不到再查找一般的文件。

```

69 \cs_new_protected:Npn \ctex_scheme_input:n #1
70   {
71     \ctex_push_file:
72       \tl_if_exist:NTF \c__ctex_class_tl
73         {
74           \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
75             { \file_input:n { ctex-scheme- #1 .def } }
76         }
77       { \file_input:n { ctex-scheme- #1 .def } }
78     \ctex_pop_file:
79   }
80 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

\g\_\_ctex\_section\_depth\_int 若大于3,则 \paragraph 和 \subparagraph 标题单独占一行;若为3,则 \paragraph 单独占一行。

```

81 <!*beamer>
82 \int_new:N \g__ctex_section_depth_int
83 \int_gset:Nn \g__ctex_section_depth_int { 2 }
84 </!beamer>

85 </!ctexsize>
86 </class|ctex|ctexheading|ctexsize>
87 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

88 \msg_new:nnnn { ctex } { package-too-old }
89   { Support~package~`#1'~too~old. }
90   {

```

```

91   Please~update~an~up-to-date~version~of~the~package~`#1'\
92   using~your~TeX~package~manager~or~from~CTAN.
93   }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

94 \sys_if_output_pdf:TF
95   { \cs_new_eq:NN \ifctexpdf \if_true: }
96   { \cs_new_eq:NN \ifctexpdf \if_false: }

```

`\ctex_if_preamble:TF` 测试是否在  $\LaTeX_2\epsilon$  的导言区。在宏包内部初始为真,文档最开始位置再设置为假。注意,钩子 `\ctex_after_end_preamble:n` 在 `\AtBeginDocument` 之后执行,可以与 `\@onlypreamble` 的行为一致。

```

97 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
98 \ctex_after_end_preamble:n { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }

```

`\ctex_set_default_ccwd:Nn` 若参数#2 带长度单位,则设置它为`tl` 变量#1 的值,否则以 `\ccwd` 为单位。

```

99 \cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
100   { \tl_set:Nc #1 { \__ctex_default_ccwd_aux:n {#2} } }
101 \cs_new:Npn \__ctex_default_ccwd_aux:n #1
102   {
103     \exp_not:n {#1}
104     \exp_after:wN \__ctex_default_ccwd_aux:w
105     \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
106   }
107 \exp_last_unbraced:NNNNo
108 \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
109   { \tl_if_empty:nT {#2} { \ccwd } }

```

`\g__ctex_encoding_tl` 所有引擎下默认编码均设为 UTF-8,初始值为空,`\ProcessKeysOptions` 再判断。

```
110 \tl_new:N \g__ctex_encoding_tl
```

`\g__ctex_zhmCJK_bool` 是否使用 `zhmCJK` 宏包。

```
111 \bool_new:N \g__ctex_zhmCJK_bool
```

`\l__ctex_autoindent_tl` 保存 `autoindent` 选项的值,空值表示不自动调整首行缩进。

```
112 \tl_new:N \l__ctex_autoindent_tl
```

`\ctex_if_autoindent_touched:F` 检查 `autoindent` 选项是否被用户设置。

```
113 \cs_new_eq:NN \ctex_if_autoindent_touched:F \use:n
```

`\ctex_zhmap_case:nnn` 参数#1 是 `zhmCJK` 的内容,#2 是 `zhmetrics`。

```
114 \cs_new_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
```

`\ctex_at_end:n` 区分 `\AtEndOfClass` 和 `\AtEndOfPackage`,虽然它们的意思都是一样的。

```

<class> 115 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 116 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfPackage }

```

`\ctex_load_std_class:n` 保存传递给标准文档类的选项。使用 `\PassOptionsToClass` 是为了预防可能存在的选项冲突,选项列表展开后再传递。

```

117 <*class>
118 \cs_new_protected:Npn \ctex_load_std_class:n #1

```

```

119 {
120   \tl_const:Nn \c__ctex_class_tl {#1}
121   \exp_args:No \PassOptionsToClass
122     { \g__ctex_std_options_clist }
123     {#1}
124   \LoadClass {#1}
125 }
126 \clist_new:N \g__ctex_std_options_clist
127 </class>

```

对无效选项给出警告。

```

128 \msg_new:nnn { ctex } { invalid-option }
129   { Option~`\l_keys_key_str'~is~invalid~in~current~mode. }
130 \msg_new:nnn { ctex } { invalid-value }
131   { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_str'. }
132 \msg_new:nnn { ctex } { CJKecglue-unsupported }
133   { `experiment/CJKecglue'~is~not~supported~by~the~current~engine. }

```

对过时选项或命令给出警告。

```

\ctex_deprecated_option:nn \ctex_set_deprecated_option:n
\ctex_deprecated_command:Nn
134 \cs_new_protected:Npn \ctex_deprecated_option:n
135   { \msg_warning:nnn { ctex } { deprecated-option } }
136 \cs_new_protected:Npn \ctex_set_deprecated_option:n #1
137   {
138     \ctex_deprecated_option:n { Option~`#1'~is~set. }
139     \ctex_set:nn { option } {#1}
140   }
141 \cs_new_protected:Npn \ctex_deprecated_command:Nn #1#2
142   {
143     \msg_warning:nnee { ctex } { deprecated-command }
144     { \token_to_str:N #1 } { \exp_not:n {#2} }
145   }
146 \msg_new:nnn { ctex } { deprecated-option }
147   { Option~`\l_keys_key_str'~is~deprecated.\\ #1 }
148 \msg_new:nnn { ctex } { deprecated-command }
149   { Command~`#1'~is~deprecated.\\ #2 }
150 </class>ctex

```

`\g__ctex_font_size_int` 0 表示修改默认字体大小为五号, 1 为小四号, 大于 1 则不作修改。初始值-1 表示 zihao 选项未初始化, 会在将来根据文档类决定初值。

```

151 <*class>ctex|ctexsize
152 \int_new:N \g__ctex_font_size_int
153 \int_gset:Nn \g__ctex_font_size_int { -1 }
154 </class>ctex|ctexsize

```

## 15.2 宏包选项

```

155 <*class>style
156 \ctex_define_option:n
157   {
158 </class>style

```

`experiment/font-size-system` 选择字号系统。预设有 word 和 letterpress 两种字号系统, 用户也可以自定义。

```

159 <*class>ctex|ctexsize
160   experiment / font-size-system .tl_gset:N = \g__ctex_font_size_system_tl ,
161   experiment / font-size-system .initial:n = { word } ,
162   experiment / font-size-system .value_required:n = true ,
163 </class>ctex|ctexsize

```

```

zihao 164 <*class|ctex|ctexsize>
165     zihao .choice: ,
166     zihao .value_required:n = true ,
167     zihao / 5 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
168     zihao / -4 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
169     zihao / false .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
<ctexsize> 170 }
171 </class|ctex|ctexsize>
172 <*class|ctex>
173     c5size .code:n = { \ctex_set_deprecated_option:n { zihao = 5 } } ,
174     cs4size .code:n = { \ctex_set_deprecated_option:n { zihao = -4 } } ,
175     c5size .value_forbidden:n = true ,
176     cs4size .value_forbidden:n = true ,

```

**linespread** 行距初始值为标志 nan, 用于检查用户是否设置了 linespread 选项。

```

177     linespread .fp_set:N = \l__ctex_line_spread_fp ,
178     linespread .initial:n = { \c_nan_fp } ,
179     linespread .value_required:n = true ,

```

**autoindent** 自动调整段落的首行缩进功能。

```

180     autoindent .choice: ,
181     autoindent .default:n = { true } ,
182     autoindent / true .code:n =
183     {
184         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
185         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
186     } ,
187     autoindent / false .code:n =
188     {
189         \tl_clear:N \l__ctex_autoindent_tl
190         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
191     } ,
192     autoindent / unknown .code:n =
193     {
194         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
195         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
196     } ,

```

**indent** 仅为兼容性保留, 已过时。

```

197     indent .code:n =
198     {
199         \ctex_deprecated_option:n
200         {
201             The~functionality~has~been~removed.\\
202             It's~better~to~set~the~heading~styles~via~`afterindent'~option.
203         }
204     } ,
205     indent .value_forbidden:n = true ,
206     noindent .code:n =
207     {
208         \ctex_deprecated_option:n
209         {
210             The~functionality~has~been~removed.\\
211             It's~better~to~set~the~heading~styles~via~`afterindent'~option.
212         }
213     } ,
214     noindent .value_forbidden:n = true ,

```

**GBK** 文档编码, 默认为 UTF-8。  
**UTF8**

```

215     GBK .code:n =

```

```

216     {
217     \sys_if_engine_pdftex:TF
218     { \tl_gset:Nn \g__ctex_encoding_tl { GBK } }
219     {
220     \msg_warning:nn { ctex } { invalid-option }
221     \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
222     }
223     } ,
224     UTF8 .code:n = { \tl_gset:Nn \g__ctex_encoding_tl { UTF8 } } ,
225     GBK .value_forbidden:n = true ,
226     UTF8 .value_forbidden:n = true ,

```

**fontset** 初始值为空。若用户未指定, 则根据操作系统载入对应字体配置, 可以区分 Windows、macOS 和其他。

```

227     fontset .tl_gset:N = \g__ctex_fontset_tl ,
228     nofonts .code:n = { \ctex_set_deprecated_option:n { fontset = none } } ,
229     adobefonts .code:n = { \ctex_set_deprecated_option:n { fontset = adobe } } ,
230     winfonts .code:n = { \ctex_set_deprecated_option:n { fontset = windows } } ,
231     nofonts .value_forbidden:n = true ,
232     winfonts .value_forbidden:n = true ,
233     adobefonts .value_forbidden:n = true ,

```

```

zhmap 234     zhmap .choice: ,
235     zhmap .default:n = { true } ,
236     zhmap / zhmCJK .code:n =
237     {
238     \bool_gset_true:N \g__ctex_zhmCJK_bool
239     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
240     } ,
241     zhmap / true .code:n =
242     {
243     \bool_gset_false:N \g__ctex_zhmCJK_bool
244     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
245     } ,
246     zhmap / false .code:n =
247     {
248     \bool_gset_false:N \g__ctex_zhmCJK_bool
249     \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
250     } ,
251     nozhmap .code:n =
252     { \ctex_set_deprecated_option:n { zhmap = false } } ,
253     nozhmap .value_forbidden:n = true ,

```

**punct** 设置标点符号输出格式。

```

254     punct .tl_set:N = \l__ctex_punct_tl ,
255     punct .default:n = { quanjiao } ,
256     punct .initial:n = { quanjiao } ,
257     nopunct .code:n = \ctex_set_deprecated_option:n { punct = plain } ,
258     nopunct .value_forbidden:n = true ,

```

```

space 259     space .choices:mn =
260     { true , auto , false }
261     { \ctex_at_end:n { \ctex_set:n { space = #1 } } } ,
262     space .default:n = { true } ,
263     nospace .code:n = { \ctex_deprecated_option:mn { space = false } } ,
264     nospace .value_forbidden:n = true ,

```

**heading** 265 heading .bool\_set:N = \l\_\_ctex\_heading\_bool ,

```

266 </class|ctex>
267 <*class|ctex|ctexheading>

```

```

sub3section 268 <!*beamer>
sub4section 269   sub3section .code:n =
270     { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
271   sub4section .code:n =
272     { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
273   sub3section .value_forbidden:n = true ,
274   sub4section .value_forbidden:n = true ,
275 </!beamer>

```

```

scheme 276   scheme .tl_set:N = \l__ctex_scheme_tl ,
277 <*ctexheading>
278   scheme .default:n = { plain } ,
279   scheme .initial:n = { plain }
280 }
281 </ctexheading>
282 <*!ctexheading>
283   scheme .default:n = { chinese } ,
284   scheme .initial:n = { chinese } ,
285 </!ctexheading>

286 </class|ctex|ctexheading>
287 <*class|ctex>

```

**cap** 和 **nocap** 是过时选项。

```

288   cap .code:n = { \ctex_set_deprecated_option:n { scheme = chinese } } ,
289   nocap .code:n = { \ctex_set_deprecated_option:n { scheme = plain } } ,
290   cap .value_forbidden:n = true ,
291   nocap .value_forbidden:n = true ,

```

以下三项都是过时的兼容选项, 它们会载入有关宏包。

```

fntef 292   fntef .code:n =
293     {
294       \sys_if_engine_xetex:TF
295       {
296         \ctex_deprecated_option:n { `xeCJKfntef'~package~is~loaded. }
297         \ctex_at_end:n { \RequirePackage { xeCJKfntef } }
298       }
299       {
300         \sys_if_engine_pdftex:TF
301         {
302           \ctex_deprecated_option:n { `CJKfntef'~package~is~loaded. }
303           \ctex_at_end:n { \RequirePackage { CJKfntef } }
304         }
305         {
306           \ctex_deprecated_option:n
307             { Furthermore,~option~`fntef'~is~invalid~in~current~mode. }
308         }
309       }
310     } ,

```

```

fancyhdr 311   fancyhdr .code:n =
312     {
313       \ctex_deprecated_option:n { `fancyhdr'~package~is~loaded. }
314       \ctex_at_end:n { \RequirePackage { fancyhdr } }
315     } ,

```

```

hyperref 316   hyperref .code:n =
317     {
318       \ctex_deprecated_option:n { `hyperref'~package~will~be~loaded. }
319       \ctex_at_end:n
320       {
321         \cs_if_exist:NF \hypersetup
322         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
323       }

```

```

324     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
325   } ,
326 }

327 </class>ctex>
328 <*class>ctex|ctexsize>

```

10pt 使 `ctex` 和 `ctexsize` 可以接受文档类的全局选项, 不修改默认字体大小。在文档类下还将参数  
 11pt 传给标准文档类。  
 12pt

```

329 \tl_clear_new:N \l__ctex_tmp_tl
330 \clist_map_inline:nn
331 {
332   10pt , 11pt , 12pt ,
333   8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
334 }
335 {
336   \tl_put_right:Nn \l__ctex_tmp_tl
337     {
338       #1 .code:n =
339 <*class>
340       { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
341 </class>
342 <*class>
343       {
344         \int_gset:Nn \g__ctex_font_size_int { 2 }
345         \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
346       } ,
347 </class>
348     #1 .value_forbidden:n = true ,
349   }
350 }
351 \exp_args:No \ctex_define_option:n { \l__ctex_tmp_tl }
352 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

353 <*class>
354 \ctex_define_option:n
355 {
356   unknown .code:n =
357     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
358 }
359 </class>

```

载入选项配置文件。

```

<|ctexsize> 360 \ctex_file_input:n { ctexopts.cfg }
361 </class>ctex|ctexsize>

```

处理宏包选项。

```

362 <*class>style>
363 \cs_if_exist:NTF \ProcessKeyOptions
364 { \ProcessKeyOptions [ ctex / option ] }
365 {
366   \RequirePackage { l3keys2e }
367   \ProcessKeysOptions { ctex / option }
368 }
369 </class>style>

```

pdfL<sup>A</sup>T<sub>E</sub>X 下, 如果没有显式指定编码为 UTF8, 则给出警告信息。

```

370 <*class>ctex>
371 \msg_new:nnn { ctex } { pdftex-utf8 }

```

```

372 { UTF8~will~be~used~as~the~default~encoding. }
373 \tl_if_empty:NT \g__ctex_encoding_tl
374 {
375   \sys_if_engine_pdfTeX:T
376   { \msg_warning:nn { ctex } { pdfTeX-utf8 } }
377   \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
378 }
379 </class>ctex>
380 <*class>

```

五号字使用标准文档类的10pt 字体大小设置,小四号字则使用12pt。

```

381 \int_case:nn { \g__ctex_font_size_int }
382 {
383   { 0 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt } }
384   { 1 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt } }
385 }

```

载入标准文档类。

```

386 <*article>
387 \ctex_load_std_class:n { article }
388 </article>
389 <*book>
390 \ctex_load_std_class:n { book }
391 </book>
392 <*report>
393 \ctex_load_std_class:n { report }
394 </report>
395 <*beamer>
396 \ctex_load_std_class:n { beamer }
397 </beamer>
398 </class>

```

## 15.3 特定引擎支持与设置

### 15.3.1 ctexbackend.cfg

对于 Xe<sub>La</sub>TeX/pdf<sub>La</sub>TeX/Lua<sub>La</sub>TeX 等默认直接输出 PDF 的编译方式,用户无需为涉及驱动的宏包指定驱动选项。对于 <sub>La</sub>TeX 和 up<sub>La</sub>TeX 等默认不直接输出 PDF 的编译方式,用户则需要指定驱动选项。

由于历史遗留问题,在使用 <sub>La</sub>TeX 或 up<sub>La</sub>TeX 等编译时,大多数涉及驱动的宏包选定的默认输出驱动都是 Dvips。考虑当前实际使用频率,以及考虑到 C<sub>T</sub>EX 宏集对中文支持的默认方式,我们在用户使用 C<sub>T</sub>EX 系列文档类时,将默认的输出驱动改为 DVIPDFM<sub>X</sub>。

具体来说,如果 dvips, dvipdfmx, dvisvgm 等驱动没有在文档类的全局选项中被明确指定,我们就在 \@classoptionslist 开头加入 dvipdfmx。

本段代码只在 ctexart 等文档类开头载入,不在 ctex 中使用。并且需要放在 expl3 之前载入,保证它载入正确的 backend 文件。

```

399 <*backend>
400 \begingroup
401 \expandafter\ifx\csname Umathchardef\endcsname\relax
402 \else\expandafter\endgroup\expandafter\endinput\fi
403 \ifodd
404   \expandafter\ifx\csname pdfoutput\endcsname\relax
405   \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
406   \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
407 \def\x#1{%

```

```

408     \if\relax\detokenize{#1}\relax
409     \gdef\@classoptionslist{dvipdfmx}%
410     \else
411     \let\CTEX@add\@ne
412     \@tfor\x:={dvips}{dvipdfmx}{dvisvgm}\do{%
413       \expandafter\in@\expandafter{\expandafter,\x,}{, #1,}%
414       \ifin@ \let\CTEX@add\tw@ \@break@tfor \fi}%
415     \ifodd\CTEX@add \gdef\@classoptionslist{dvipdfmx, #1}\fi
416     \fi}
417     \expandafter\x\expandafter{\@classoptionslist}
418   \fi
419 \endgroup
420 </backend>

```

### 15.3.2 ctex-engine-pdfTeX.def

`\ctex_set_zhmap:n` 设置 $\text{upTeX}$  字体映射, 同时作用于 `\AtBeginDvi` 与 `\AtBeginShipoutFirst`。该宏对 $\text{pdfTeX}$  和 $\text{upTeX}$  均有用。`\AtBeginDvi` 直接将 `\special` 保存到盒子中, `\AtBeginShipoutFirst` 是保存到宏中, 并且不展开参数。

可以使用 $\text{LATEX}$  2020/10/01 的钩子机制来统一设置。

```

421 <*pdfTeX|upTeX|apTeX>
422 \cs_new_protected:Npn \ctex_set_zhmap:n
423   { \tl_gput_right:Ne \g__ctex_zhmap_tl }
424 \cs_new_protected:Npn \ctex_use_zhmap:
425   { \tl_use:N \g__ctex_zhmap_tl }
426 \cs_if_exist:NTF \ctex_gadd_ltxhook:nn
427   {
428     \cs_new_protected:Npn \ctex_at_shipout_first:n
429       { \ctex_gadd_ltxhook:nn { shipout/firstpage } }
430     \ctex_at_shipout_first:n { \ctex_use_zhmap: }
431   }
432   {
433     \cs_new_protected:Npn \ctex_add_dvi_zhmap:
434       { \AtBeginDvi { \ctex_use_zhmap: } }
435     \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
436     \ctex_at_end_package:nn { atbegshi }
437     {
438       \cs_new_protected:Npn \ctex_at_shipout_first:n
439         { \AtBeginShipoutFirst }
440       \ctex_at_shipout_first:n { \ctex_use_zhmap: }
441       \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
442     }
443   }
444 \tl_new:N \g__ctex_zhmap_tl
445 \@onlypreamble \ctex_set_zhmap:n
446 </pdfTeX|upTeX|apTeX>
447 <*pdfTeX>

```

`\c__ctex_cmap_encoding_seq` 需要加上 CMap 的 CJK 字体编码。

```

448 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
449   { C19 , C10 , C00 , C09 , C40 , C60 }

```

`\ctex_family_cmap:nn` 在 `\DeclareFontFamily` 的 $\langle\text{loading-settings}\rangle$  中给 CJK 字体族加上 CMap。

```

450 \cs_new_protected:Npn \ctex_family_cmap:nn #1#2
451   {
452     \cs_if_free:cF { #1 + #2 }
453     {
454       \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
455       { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }

```

```

456     }
457 }
458 \cs_generate_variant:Nn \ctex_family_cmap:nn { ee }
459 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:ee

```

`\ctex_add_cmap:n` 给#1 编码的 CJK 字体加上 CMap。

```

460 \cs_new_protected:Npn \ctex_add_cmap:n #1
461 {
462   \cs_if_free:NF \CJK@plane
463   { \exp_args:Ne \__ctex_add_cmap_auxi:n { #1 \CJK@plane } }
464 }
465 \cs_new_protected:Npn \__ctex_add_cmap_auxi:n #1
466 { \exp_args:Nc \__ctex_add_cmap_auxii:Nn { \__ctex_add_cmap_ #1 : } {#1} }
467 \cs_new_protected:Npn \__ctex_add_cmap_auxii:Nn #1#2
468 {
469   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
470   #1
471 }
472 \cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
473 {
474   \exp_args:Ne \file_get_full_name:nNTF
475   { \str_lowercase:n {#2} .cmap } \l__ctex_cmap_file_tl
476   {
477     \pdf_object_unnamed_write:ne
478     { fstream }
479     { { } { \l__ctex_cmap_file_tl } }
480     \cs_new_protected:Npx #1
481     {
482       \pdfnobluiltintounicode \tex_font:D
483       \tex_pdffontattr:D \tex_font:D
484       { /ToUnicode ~ \pdf_object_ref_last: }
485     }
486   }
487   { \cs_new_eq:NN #1 \prg_do_nothing: }
488 }
489 \tl_new:N \l__ctex_cmap_file_tl

```

`\DeclareFontFamily` 只在pdfL<sup>A</sup>T<sub>E</sub>X 下加 CMap。如 cmap 宏包被引入, 则不重复设置。

```

490 \sys_if_output_pdf:T
491 {
492   \exp_args:Nno \use:n
493   { \cs_gset_protected:Npn \DeclareFontFamily #1#2#3 }
494   {
495     \DeclareFontFamily {#1} {#2} {#3}
496     \CTEX@Family@CMap {#1} {#2}
497   }
498   \ctex_at_end_package:nn { cmap }
499   {
500     \cs_gset_eq:NN \ctex_add_cmap:n \use_none:n
501     \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn
502   }
503 }

```

首先检查选项, 决定是否载入 zhmCJK 宏包。

```

504 \bool_if:NTF \g__ctex_zhmCJK_bool
505 {
506   \exp_args:Ne \PassOptionsToPackage
507   { encoding = \g__ctex_encoding_tl }
508   { zhmCJK }
509   \RequirePackage { zhmCJK }
510 }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

511 {
512   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
513   { \RequirePackage { CJK } }
514   { \RequirePackage { CJKutf8 } }
515   \RequirePackage { CJKpunct , CJKspace }

```

`\ctex_load_zhmap:n` 载入 zhmetrics 的字体映射文件, 同时设置 `\CJKrmdefault` 等。

```

516   \cs_new_protected:Npn \ctex_load_zhmap:n #1#2#3#4
517   {
518     \tl_set:Nn \CJKrmdefault {#1}
519     \tl_set:Nn \CJKsfdefault {#2}
520     \tl_set:Nn \CJKTtdefault {#3}
521     \ctex_set_zhmap:n { \ctex_file_input:n { ctex-zhmap- #4 .tex } }
522   }
523   \@onlypreamble \ctex_load_zhmap:n
524 }

```

`\ctex_undefine_unicode_character:n` L<sup>A</sup>T<sub>E</sub>X 2022-06-01 将通常用作中文书名号的 U+3008 和 U+3009 分别定义为 `\textlangle` 和 `\textrangle`, 这个定义会被 CJKutf8 包优先使用。我们总是使用中文书名号, 需要取消 L<sup>A</sup>T<sub>E</sub>X 的定义, 作用是局部的, 不使用 `\cs_undefine:N`。

```

525 \cs_new_protected:Npn \ctex_undefine_unicode_character:n #1
526 { \cs_set_eq:cN { u8 : \codepoint_str_generate:n {#1} } \tex_undefined:D }
527 \CJKaddEncHook { UTF8 }
528 {
529   \ctex_undefine_unicode_character:n { "3008 }
530   \ctex_undefine_unicode_character:n { "3009 }
531 }

```

`\ctex_CJK_input:n` 载入 CJK 包的 .enc 和 .bdg 等文件时, 需要设置 `\endlinechar` 为 -1。

```

\CJK@input
532 \cs_new_protected:Npn \ctex_CJK_input:n #1
533 {
534   \ctex_push_file:
535     \int_set:Nn \tex_endlinechar:D { -1 }
536     \file_input:n {#1}
537   \ctex_pop_file:
538 }
539 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

`\ctex_plane_to_utfxvibe:Nn` fancyhdr 宏包的 `\nouppercase` 会将 `\uppercase` 定义为 `\relax`, 而 `\CJK@surr` 需要用它将 `\CJK@plane` 转化成大写字母, 这就造成了冲突<sup>18</sup>。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

540 \cs_if_free:NF \CJK@surr
541 {
542   \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
543   {
544     \int_set:Nn \l__ctex_tmp_int
545     { \exp_args:Ne \int_from_hex:n {#2} }
546     \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
547     { \tl_gset:Ne #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
548     {
549       \int_sub:Nn \l__ctex_tmp_int { 256 }
550       \tl_gset:Ne #1
551       {
552         \int_to_Hex:n
553         { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }

```

<sup>18</sup><https://github.com/CTeX-org/ctex-kit/issues/146>

```

554             \int_to_Hex:n
555             { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
556         }
557     }
558 }
559 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
560 }

```

\CJK@addcmap L<sup>A</sup>T<sub>E</sub>X 2021-06-01 默认载入 `glyphtounicode.tex`, 我们对 CJK 字体禁用这一内建设置。

```

561 \cs_if_free:NF \CJK@addcmap
562 {
563     \ctex_patch_cmd:Nnn \CJK@addcmap
564     { \pdffontattr \font@name }
565     {
566         \pdfnointintounicode \font@name
567         \pdffontattr \font@name
568     }
569 }

```

CJKpunct 宏包会在 `\AtBeginDocument` 的里设置标点格式为 `quanjiao`。

```

570 \AtBeginDocument
571 {
572     \use:e
573     {
574         \str_if_eq:nnF { \l__ctex_punct_tl } { quanjiao }
575         { \exp_not:N \punctstyle { \l__ctex_punct_tl } }
576     }
577 }

```

在导言区末尾更新 `\CJKfamilydefault`, 注意要在 `\CJK@envStart` 之前使用。

```

578 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

从 L<sup>A</sup>T<sub>E</sub>X 2018-04-01 开始, 传统引擎的默认编码被设置成 UTF-8, 汉字的首字节已经被设置成活动字符, 我们无需再使用 `\CJK@makeActive` 设置。 `\CJK@envStart` 的定义是

```

\def\CJK@envStart#1#2#3{
  \CJK@upperReset
  \ifCJK@lowercase@
    \CJK@lowerReset
  \fi%
  \CJK@makeActive%
  \CJK@global\let\CJK@selectFamily \CJK@selFam
  \CJK@global\let\CJK@selectEnc \CJK@selEnc%
  \def\CJK@@@enc{#2}
  \ifx\CJK@@@enc \empty
    \PackageInfo{CJK}{
      no encoding parameter given,\MessageBreak
      waiting for \protect\CJKenc\space commands}
  \else
    \CJKenc{#2}
  \fi
  \CJKfontenc{#2}{#1}
  \CJKfamily{#3}
  \def\CJK@series{f@series}
  \def\CJK@shape{f@shape}%
  \csname CJKhook\endcsname}

```

`\CJK@upperReset` 可能会有一定风险, 因此我们直到导言区末尾才使用 `\CJK@envStart`。这样可以避免将 CJK 环境内置入 `document` 环境的最里层, 最后也就不需要 `\clearpage`。 `zhmCJK` 已经提供类似功能。注意先使用 `\ctex_update_default_family:` 更新 `\CJKfamilydefault`。

```

579 \bool_if:NF \g__ctex_zhmCJK_bool

```

```

580 {
581   \str_if_eq:onTF { \g__ctex_encoding_tl } { UTF8 }
582   { \CJK@loadBinding { UTF8 } }
583   { \CJK@loadBinding { standard } }
584   \exp_args:Ne \ctex_at_end_preamble:n
585   {
586     \exp_not:N \CJK@envStart
587     { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
588     \exp_not:N \CJKtilde
589   }
590 }

```

\ctex\_auto\_ignorespaces: 保存 \CJK@@ignorespaces 的定义, 方便使用。

```

591 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces

```

\CJK@X 最新 L<sup>A</sup>T<sub>E</sub>X 中的 \MakeUppercase 等大小写转化命令被定义为 L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> 中对应的 \text\_uppercase:n 等函数。它会先将参数用 \text\_expand:n 逐步展开, 这与 CJK 包的定义不兼容。可以有多种方法解决这个冲突, 最简单直接的方法是将首字节的活动字符用 \protected 来定义, 但可能有一定的兼容性问题。保险起见, 我们采用将 \CJK@X 等内部宏转化为 \protected 宏的方式, 并将参数提前由字符转化为数字, 避免在移动参数中被进一步展开。

```

592 \cs_new:Npn \__ctex_char:N #1
593 { \exp_args:Ne \CTEX@char@n { \int_value:w `#1 } }
594 \cs_new:Npn \__ctex_char:NN #1#2
595 {
596   \token_if_eq_meaning:NNTF #2 \protect
597   { \__ctex_char_aux:NN #1 }
598   { \__ctex_char_aux:NN #1#2 }
599 }
600 \cs_new:Npn \__ctex_char_aux:NN #1#2
601 {
602   \exp_last_unbraced:Ne \CTEX@char@nn
603   {
604     { \int_value:w `#1 }
605     { \int_value:w `#2 }
606   }
607 }
608 \cs_new:Npn \__ctex_char:NNN #1#2#3
609 {
610   \token_if_eq_meaning:NNTF #2 \protect
611   { \__ctex_char_auxi:NNNN #1#3 }
612   { \__ctex_char_auxii:NNN #1#2#3 }
613 }
614 \cs_new:Npn \__ctex_char_auxi:NNNN #1#2#3#4
615 { \__ctex_char_auxii:NNN #1#2#4 }
616 \cs_new:Npn \__ctex_char_auxii:NNN #1#2#3
617 {
618   \exp_last_unbraced:Ne \CTEX@char@nnn
619   {
620     { \int_value:w `#1 }
621     { \int_value:w `#2 }
622     { \int_value:w `#3 }
623   }
624 }
625 \cs_new:Npn \__ctex_char:NNNN #1#2#3#4
626 {
627   \token_if_eq_meaning:NNTF #2 \protect
628   { \__ctex_char_auxi:NNNN #1#3 }
629   { \__ctex_char_auxii:NNNN #1#2#3#4 }
630 }
631 \cs_new:Npn \__ctex_char_auxi:NNNN #1#2#3#4#5
632 { \__ctex_char_auxii:NNNN #1#2#3#5 }
633 \cs_new:Npn \__ctex_char_auxii:NNNN #1#2#3#4

```

```

634 {
635   \exp_last_unbraced:Ne \CTEX@char@nnnn
636   {
637     { \int_value:w `#1 }
638     { \int_value:w `#2 }
639     { \int_value:w `#3 }
640     { \int_value:w `#4 }
641   }
642 }
643 \cs_gset_eq:NN \CJK@X   \__ctex_char:N
644 \cs_gset_eq:NN \CJK@XX  \__ctex_char:NN
645 \cs_gset_eq:NN \CJK@XXX \__ctex_char:NNN
646 \cs_gset_eq:NN \CJK@XXXX \__ctex_char:NNNN

```

CJKutf8 包优先使用 `\u8:xxx/\u8:xxxx` 的 L<sup>A</sup>T<sub>E</sub>X 定义。

```

647 \str_if_eq:onF { \g__ctex_encoding_tl } { GBK }
648 {
649   \exp_args:Nne \use:n
650   { \cs_gset:Npn \__ctex_char_aux:NN #1#2 }
651   {
652     \exp_not:N \cs_if_exist_use:cF
653     { u8: \exp_not:N \tl_to_str:n { #1#2 } }
654     { \exp_not:o { \__ctex_char_aux:NN {#1} {#2} } }
655   }
656   \exp_args:Nne \use:n
657   { \cs_gset:Npn \__ctex_char_auxii:NNN #1#2#3 }
658   {
659     \exp_not:N \cs_if_exist_use:cF
660     { u8: \exp_not:N \tl_to_str:n { #1#2#3 } }
661     { \exp_not:o { \__ctex_char_auxii:NNN {#1} {#2} {#3} } }
662   }
663   \exp_args:Nne \use:n
664   { \cs_gset:Npn \__ctex_char_auxiii:NNNN #1#2#3#4 }
665   {
666     \exp_not:N \cs_if_exist_use:cF
667     { u8: \exp_not:N \tl_to_str:n { #1#2#3#4 } }
668     { \exp_not:o { \__ctex_char_auxiii:NNNN {#1} {#2} {#3} {#4} } }
669   }
670 }

```

`\mule@arg` 处理文档被 `extconv` 等工具预处理过的情况, 只考虑最简单的情形。

```

671 \group_begin:
672 \char_set_catcode_active:n { "7F }
673 \cs_new:Npn \__ctex_char_preproc:w #1 ^^7f #2 ^^7f
674 { \exp_args:No \CTEX@char@nn { \int_value:w `#1 } {#2} }
675 \cs_gset_eq:NN \mule@arg \__ctex_char_preproc:w
676 \group_end:
677 \exp_args:Ne \ctex_at_end:n
678 {
679   \char_set_catcode:nn
680   { \int_value:w "7F }
681   { \char_value_catcode:n { "7F } }
682 }

```

额外加入 `\mode_leave_vertical:` 是为了解决某些历史遗留问题。

```

\CTEX@char@n  \CTEX@char@nn  \CTEX@char@nnn  \CTEX@char@nnnn
\CTEX@char@n 683 \cs_new_protected:Npn \CTEX@char@n #1
\CTEX@char@nn 684 {
\CTEX@char@nnn 685   \mode_leave_vertical:
\CTEX@char@nnnn 686   \use:c { CJK@ #1 }
687   \CJK@ignorespaces
688 }
689 \cs_new_protected:Npn \CTEX@char@nn #1#2
690 {

```

```

691 \mode_leave_vertical:
692 \use:c { CJK@ #1 } {#2}
693 \CJK@ignorespaces
694 }
695 \cs_new_protected:Npn \CTEX@char@nnn #1#2#3
696 {
697 \mode_leave_vertical:
698 \use:c { CJK@ #1 } {#2} {#3}
699 \CJK@ignorespaces
700 }
701 \cs_new_protected:Npn \CTEX@char@nnnn #1#2#3#4
702 {
703 \mode_leave_vertical:
704 \use:c { CJK@ #1 } {#2} {#3} {#4}
705 \CJK@ignorespaces
706 }

```

\CTEX@char@nn@n 以下给 \CTEX@char@nn 等定义在 \text\_expand:n 中的等价形式, 将它们的参数用只有一个参数的 \CTEX@char@nn@n 等包装起来, 并将 \CTEX@char@nn@n 等放入排除列表中, 避免可能的大小写转化。

```

707 \cs_new_protected:Npn \CTEX@char@nn@n #1 { \CTEX@char@nn #1 }
708 \cs_new_protected:Npn \CTEX@char@nnn@n #1 { \CTEX@char@nnn #1 }
709 \cs_new_protected:Npn \CTEX@char@nnnn@n #1 { \CTEX@char@nnnn #1 }
710 \cs_new:Npn \__ctex_char_wrap:nn #1#2
711 { \CTEX@char@nn@n { {#1} {#2} } }
712 \cs_new:Npn \__ctex_char_wrap:nnn #1#2#3
713 { \CTEX@char@nnn@n { {#1} {#2} {#3} } }
714 \cs_new:Npn \__ctex_char_wrap:nnnn #1#2#3#4
715 { \CTEX@char@nnnn@n { {#1} {#2} {#3} {#4} } }
716 \tl_put_right:Nn \l_text_case_exclude_arg_tl
717 {
718 \CTEX@char@n
719 \CTEX@char@nn@n
720 \CTEX@char@nnn@n
721 \CTEX@char@nnnn@n
722 }
723 \text_declare_expand_equivalent:Nn \CTEX@char@nn { \__ctex_char_wrap:nn }
724 \text_declare_expand_equivalent:Nn \CTEX@char@nnn { \__ctex_char_wrap:nnn }
725 \text_declare_expand_equivalent:Nn \CTEX@char@nnnn { \__ctex_char_wrap:nnnn }

```

以下给 \CTEX@char@nn@n 等定义在 \text\_purify:n 中的等价形式, 将参数中的数字转化回字符。

```

726 \cs_new:Npn \__ctex_char_raw:n #1
727 { \char_generate:nn {#1} { 12 } }
728 \cs_new:Npn \__ctex_char_raw:nn #1#2
729 {
730 \char_generate:nn {#1} { 12 }
731 \char_generate:nn {#2} { 12 }
732 }
733 \cs_new:Npn \__ctex_char_raw:nnn #1#2#3
734 {
735 \char_generate:nn {#1} { 12 }
736 \char_generate:nn {#2} { 12 }
737 \char_generate:nn {#3} { 12 }
738 }
739 \cs_new:Npn \__ctex_char_raw:nnnn #1#2#3#4
740 {
741 \char_generate:nn {#1} { 12 }
742 \char_generate:nn {#2} { 12 }
743 \char_generate:nn {#3} { 12 }
744 \char_generate:nn {#4} { 12 }
745 }
746 \cs_new:Npn \__ctex_char_raw_nn:n #1 { \use:e { \__ctex_char_raw:nn #1 } }
747 \cs_new:Npn \__ctex_char_raw_nnn:n #1 { \use:e { \__ctex_char_raw:nnn #1 } }

```

```

748 \cs_new:Npn \__ctex_char_raw_nnnn:n #1 { \use:e { \__ctex_char_raw:nnnn #1 } }
749 \text_declare_purify_equivalent:Nn \CTEX@char@n { \__ctex_char_raw:n }
750 \text_declare_purify_equivalent:Nn \CTEX@char@nn@n { \__ctex_char_raw:nn:n }
751 \text_declare_purify_equivalent:Nn \CTEX@char@nnn@n { \__ctex_char_raw:nnn:n }
752 \text_declare_purify_equivalent:Nn \CTEX@char@nnnn@n { \__ctex_char_raw:nnnn:n }

```

`\ctex_ignorespaces_case:N` 设置忽略空格的的方式。根据 `space` 选项的值重定义 `\CJK@ignorespaces`，并保存起来供 `\ctex_set_ignorespaces:` `\CJKhook` 备用。

```

753 \cs_new_protected:Npn \ctex_ignorespaces_case:N #1
754 {
755   \cs_set_protected:Npn \ctex_set_ignorespaces:
756     { \cs_set_eq:NN \CJK@ignorespaces #1 }
757   \ctex_set_ignorespaces:
758 }
759 \cs_new_protected:Npn \ctex_set_ignorespaces:
760 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }

```

`\CJKhook` `CJK` 和 `CJK*` 环境都会重新定义 `\CJK@ignorespaces`。我们在 `CJK` 宏包提供的 `\CJKhook` 里重新设置它，让这两个环境忽略空格的方式都受 `space` 选项的控制。这对 `zhmCJK` 是必要的。

```

761 \tl_if_exist:NF \CJKhook { \tl_new:N \CJKhook }
762 \tl_gput_right:Nn \CJKhook { \ctex_set_ignorespaces: }

```

`\ctex_punct_set:n` 设置 `CJK` 族对应到实际的字体。`#1` 是 `fontset` 的名字。

```

763 \cs_new_protected:Npn \ctex_punct_set:n #1
764 {
765   \clist_map_inline:Nn \c__ctex_punct_family_clist
766     {
767       \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
768         {
769           \cs_set_eq:cc
770             { CJKpunct@ ##1 @spaces }
771             { c__ctex_ #1 ##1 _punct_spaces_tl }
772         }
773     }
774 }
775 \clist_const:Nn \c__ctex_punct_family_clist
776 {
777   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
778   zhsongb , zhheil , zhheib , zhyoub ,
779   zhyahei , zhyaheib , zhpfb , zhpfb
780 }

```

`\ctex_punct_map_family:nn` `CJK` 族#1 使用族#2 的边界信息。

```

781 \cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
782 {
783   \cs_if_free:cF { CJKpunct@ #2 @spaces }
784     { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
785 }

```

`\ctex_punct_map_bfseries:nn` `CJK` 族#1 的 `\bfseries` 使用族#2 的边界信息。

```

786 \cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
787 {
788   \clist_map_inline:nn {#1}
789     {
790       \ctex_punct_map_series:nnn { #1 } { b } {#2}
791       \ctex_punct_map_series:nnn { #1 } { bx } {#2}
792     }
793 }
794 \cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
795 {

```

```

796 \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
797 \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
798 \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
799 \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
800 \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
801 \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
802 }

```

`\ctex_punct_map_itshape:nn` CJK 族#1 的 `\itshape` 使用族#2 的边界信息。

```

803 \cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
804 {
805   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
806   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
807   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
808   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
809   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
810   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
811 }

```

`\ctex_punct_space:nn` 定义标点的边界信息。

`\ctexspadef`

```

812 \cs_new_protected:Npn \ctex_punct_space:nn #1#2
813 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
814 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```
815 \ctex_file_input:n { ctexspa.def }
```

816 [⟨pdfTeX⟩](#)

### 15.3.3 ctex-engine-xetex.def

817 [⟨\\*xetex⟩](#)

```

818 \RequirePackage { xeCJK }
819 \exp_args:Ne \xeCJKsetup
820 {
821   LoadFandol = false ,
822   PunctStyle = \l__ctex_punct_tl
823 }

```

最新版本的 `fontspec` 默认对 `\rmfamily` 和 `\sffamily` 设置 `Ligatures=TeX`，对 `\ttfamily` 设置 `WordSpace={1,0,0}` 和 `PunctuationSpace=WordSpace`。

```

824 \@ifpackagelater { fontspec } { 2014/05/25 } { }
825 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

826 [⟨xetex⟩](#)

### 15.3.4 ctex-engine-luatex.def

827 [⟨\\*luatex⟩](#)

LuaTeX-ja 为了兼容 p $\text{\LaTeX}$  的使用习惯，对  $\text{\LaTeX}2_{\epsilon}$  的 NFSS 作了不少修改和扩充，这对于简体中文用户来说不是必要的。我们在这里禁用它。

```

828 \msg_new:nnn { ctex } { luatexja-loaded }
829 {
830   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
831   Loading~file~`#1'~will~abort!
832 }
833 \@ifpackageloaded { luatexja }
834 { \msg_critical:nne { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
835 {

```

```

836 \ctex_at_begin_package:nn { luatexja }
837   { \msg_redirect_name:nnn { ctexhook } { disable-package } { info } }
838 \ctex_at_end_package:nn { luatexja }
839   { \msg_redirect_name:nnn { ctexhook } { disable-package } { } }
840 \ctex_disable_package:n { ltj-latex }
841 }

842 \RequirePackage { luatexja }
843 \@ifpackagelater { luatexja } { 2020/04/12 } { }
844 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }

845 \RequirePackage { fontspec }
846 \@ifpackagelater { fontspec } { 2020/02/21 } { }
847 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

引擎文件是通过 `\ctex_file_input:n` 载入的, 其中的 `\catcodetable` 机制会完整恢复文件载入之前的 `\catcode` 状态, 在引擎文件中的 `\catcode` 设置都无效。因此, 对 `\ltjlineendcomment` 的设置要放到文件之外进行。

```

848 \ctex_at_end:n { \char_set_catcode_comment:n { \ltjlineendcomment } }

```

### 15.3.4.1 LuaTeX-ja 的默认设置

```

849 \ExplSyntaxOff

```

以下设置抄录自 `l1tjdefs.sty`, 略有改动。

U+2460–U+24FF (Enclosed Alphanumerics) 原属于字符范围 6, 是 JAchar, 我们把它们归入字符范围 3, 改成 ALchar。

```

850 \ltjdefcharrange{1}{"80-"36F, "1E00-"1EFF}
851 \ltjdefcharrange{2}{"370-"4FF, "1F00-"1FFF}
852 \ltjdefcharrange{3}{%
853   "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
854 \ltjdefcharrange{4}{%
855   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
856   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
857   "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFFF, "1B170-"1F0FF,
858   "1F300-"1FFFF, "2000-"206F}
859 \ltjdefcharrange{5}{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
860 \ltjdefcharrange{6}{%
861   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
862   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,
863   "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
864 \ltjdefcharrange{7}{%
865   "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
866   "A960-"A97F, "AC00-"D7FF}
867 \ltjdefcharrange{8}{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}

```

将间隔号、引号、破折号等中西文公用的标点符号归入字符范围 9, 将他们设置为 JAchar。

```

868 \ltjdefcharrange{9}{%
869   "00B7, "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

LuaTeX-ja 默认把字符范围 2 和 3 设置为 JAchar, 我们这里把它们都改成 ALchar。

```

870 \ltjsetparameter{jacharrange={-1, -2, -3, -4, -5, +6, +7, -8, +9}}
871 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 2020/08/08 之前的 `ltj-latex.sty`。自 2020/08/08 开始, LuaTeX-ja 引入新的缓存机制, 此段设置被整合进 `luatexja.lua`。

```

872 \@ifpackagelater{luatexja}{2020/08/08}
873 { \ltjsetparameter { autospacing, autoxspacing, differentjfm = paverage } }
874 {
875   \directlua{
876     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
877     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
878   }

```

```

879 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
880   xkanjiskip=.25\zw plus 1pt minus 1pt,
881   autospacing, autoxspacing, jacharrange={-1},
882   yalbaselineshift=\z@, yjabaselineshift=\z@,
883   jcharwidowpenalty=500, differentjfm=paverage
884 }
885 }
886 \ExplSyntaxOn

```

### 15.3.4.2 LuaTeX-ja 的补丁

```
887 <@@=ctex_ltj>
```

\CTEX@alchar 分组中的字符都是 ALchar 类。

```

\CTEX@beginallalchar
\CTEX@endallalchar
888 \cs_new_protected:Npn \CTEX@alchar #1
889 { \CTEX@beginallalchar #1 \CTEX@endallalchar }
890 \cs_new_protected:Npn \CTEX@beginallalchar
891 {
892   \group_begin:
893   \ctex_ltj_zero_globaldefs:
894   \ltj@allalchar
895 }
896 \cs_new_protected:Npn \CTEX@endallalchar
897 { \group_end: }

```

\CTEX@chardef@text@cmd 补丁 \chardef@text@cmd, 应用于 \DeclareTextSymbol, 使其定义的符号都是 ALchar。

```

898 \cs_new_protected:Npn \CTEX@chardef@text@cmd #1
899 {
900   \cs_set_eq:NN \@ifdefinable \@ifdefinable
901   \tl_set:Nn \l__ctex_ltj_cmd_tl {#1}
902   \tex_afterassignment:D \__ctex_ltj_chardef_text_cmd:
903   \tex_chardef:D #1
904 }
905 \tl_new:N \l__ctex_ltj_cmd_tl
906 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd:
907 { \exp_after:wN \__ctex_ltj_chardef_text_cmd_aux:N \l__ctex_ltj_cmd_tl }
908 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd_aux:N #1
909 {
910   \int_compare:nNnF {#1} < { "80 }
911   { \cs_set_protected:Npx #1 { \CTEX@alchar { \tex_Uchar:D #1 } } }
912 }

```

\CTEX@text@composite@x \@text@composite@x 的重定义, 应用于 \DeclareUnicodeComposite 等。

```

913 \cs_new_protected:Npn \CTEX@text@composite@x #1#2
914 {
915   \CTEX@beginallalchar
916   \cs_if_exist_use:NF #1 {#2}
917   \CTEX@endallalchar
918 }

```

\CTEX@add@unicode@accent \add@unicode@accent 的重定义, 应用于 \DeclareUnicodeAccent。

```

919 \cs_new_protected:Npx \CTEX@add@unicode@accent #1#2
920 {
921   \CTEX@beginallalchar
922   \exp_not:N \tl_if_blank:nTF {#2} { \tex_Uchar:D "A0 ~ } {#2}
923   \exp_not:N \tex_Uchar:D \tex_numexpr:D #1 \scan_stop:
924   \CTEX@endallalchar
925 }

```

\CTEX@patch@text@cmd 单独补丁由 \DeclareTextCommand 定义的命令。

```
926 \cs_new_protected:Npn \CTEX@patch@text@cmd #1
```

```

927 {
928   \exp_args:Nnc \__ctex_ltj_patch_text_cmd:NN #1
929   { \UnicodeEncodingName \token_to_str:N #1 }
930 }
931 \cs_new_protected:Npn \__ctex_ltj_patch_text_cmd:NN #1#2
932 {
933   \cs_set_eq:NN \CTEX@textcmd #2
934   \ctex_preto_cmd:NnnTF \CTEX@textcmd
935   { \ExplSyntaxOff \makeatletter }
936   { \CTEX@beginallalchar }
937   {
938     \ctex_appto_cmd:NnnTF \CTEX@textcmd
939     { \ExplSyntaxOff \makeatletter }
940     { \CTEX@endallalchar }
941     { \cs_set_eq:NN #2 \CTEX@textcmd }
942     { \ctex_patch_failure:N #1 }
943   }
944   { \ctex_patch_failure:N #1 }
945 }

```

`\CTEX@patch@tunec` 重新载入 `tunec.def`, 使补丁生效。

```

946 \cs_new_protected:Npn \CTEX@patch@tunec
947 {
948   \cs_set_eq:NN \chardef@text@cmd \CTEX@chardef@text@cmd
949   \ctex_file_input:n { tuenc.def }
950   \cs_set_eq:NN \@text@composite@x \CTEX@text@composite@x
951   \cs_set_eq:NN \add@unicode@accent \CTEX@add@unicode@accent
952   \CTEX@patch@text@cmd \textasteriskcentered
953 }
954 \@ifpackageloaded { xunicode }
955 { }
956 { \CTEX@patch@tunec }

```

在  $\text{\LaTeX}$  下, `LuaTeX-ja` 对 `fontspec`、`xunicode`、`unicode-math` 和 `listings` 打了补丁。其中前三个是把 `\char` 换成 `\ltjalchar`, 确保字符是 `ALchar` 类。我们这里用 `xunicode-addon` 来处理 `xunicode`。

```

957 \ctex_at_end_package:n { xunicode }
958 {
959   \RequirePackage { xunicode-addon }
960   \AtBeginUTFCommand { \CTEX@beginallalchar }
961   \AtEndUTFCommand { \CTEX@endallalchar }
962 }

```

对 `listings` 的补丁是让代码环境支持 `JAchar` 类。

```

963 \ctex_at_end_package:n { listings }
964 { \RequirePackage { lltjp-listings } }

```

### 15.3.4.3 Lua 函数

直接用 `token.set_lua` 定义, 不是传统意义上的  $\text{\TeX}$  宏。

```

965 \group_begin:
966 \char_set_catcode_space:n { 32 }
967 \lua_now:e
968 {

```

`ctex.newluacmd` 定义新的 Lua 函数。

```

969   ctex = ctex or { }
970   local ctex = ctex
971   local functions = lua.get_functions_table()

```

```

972 local new_luafunction = luatexbase.new_luafunction
973 local create, set_lua = token.create, token.set_lua
974 local lua_cmds = {
975     lua_call          = true ,
976     lua_expandable_call = true ,
977 }
978 local newluacmd = function (name, func, ...)
979     local tok = create(name)
980     local id = lua_cmds[tok.cmdname] and tok.index
981     local id = id or new_luafunction(name)
982     set_lua(name, id, ...)
983     functions[id] = func
984 end
985 ctex.newluacmd = newluacmd

986 local ltjfont = luatexja.jfont
987 local getattribute = tex.getattribute
988 local tex_set, sprint = tex.set, tex.sprint
989 local scan_int, scan_arg = token.scan_int, token.scan_argument

```

`\ctex_ltj_add_kyenc:n` 保存 jfont 的编码,用于判断。

```

990 newluacmd("ctex_ltj_add_kyenc:n", ltjfont.add_kyenc_list, "global", "protected")

```

`\ctex_ltj_is_kenc:n` 判断编码是否属于 jfont。

```

991 newluacmd("ctex_ltj_is_kenc:n", ltjfont.is_kenc, "global")

```

`\ctex_ltj_patch_external_font:n` 若对字体的定义完全相同,则它们有相同的 font.id。因此如果字形是由 NFSS 的替换机制定义的,它们就有相同的 font.id。print\_aftl\_address 函数的定义是

```

function luatexja.jfont.print_aftl_address()
    return ';ltjaltfont' .. tostring(aftl_base):sub(8)
end

```

主要目的是,如果当前字形有替代字体,则往字形的定义中加入一些标志,确保 font.id 唯一。

```

992 newluacmd("ctex_ltj_patch_external_font:n", function ()
993     local s = scan_arg()
994     local is_braced, is_quoted
995     if s:sub(1,1) == '{' and s:sub(-1)=='}' then
996         is_braced = true; s = s:sub(2,-2)
997     end
998     if s:sub(1,1) == '"' and s:sub(-1) == '"' then
999         is_quoted = true; s = s:sub(2,-2)
1000    end
1001    s = s .. ltjfont.print_aftl_address()
1002    if is_braced then s = '{' .. s .. '}'
1003    elseif is_quoted then s = '"' .. s .. '"'
1004    end
1005    sprint(-2, s)
1006    end, "global")

```

`\ctex_ltj_set_alt_font:nmmn` #1 和#2 分别是字符区间的首末,#3 是基础字体,#4 是替代字体。

```

1007 newluacmd("ctex_ltj_set_alt_font:nmmn", function ()
1008     local b = tonumber(scan_arg())
1009     local e = tonumber(scan_arg())
1010     local alt = scan_arg()
1011     local base = scan_arg()
1012     ltjfont.set_alt_font_latex(b, e, alt, base)
1013     end, "global", "protected")

```

`\ctex_ltj_clear_alt_font:n` 清除#1 的替代字体。

```

1014 newluacmd("ctex_ltj_clear_alt_font:n", function ()

```

```

1015     local base = scan_arg()
1016     ltjfont.clear_alt_font_latex(base)
1017     end, "global", "protected")

```

`\ctex_ltj_pickup_alt_font:n` 定义#1 的替代字体, #2 是字体大小。  
会在内部执行 `\ltj@pickup@altfont@auxy` 和 `\ltj@pickup@altfont@copy`。

```

1018     newluacmd("ctex_ltj_pickup_alt_font:n", function ()
1019         local base = scan_arg()
1020         local size = scan_arg()
1021         ltjfont.output_alt_font_cmd("y", base)
1022         ltjfont.pickup_alt_font_a(size)
1023     end, "global", "protected")

```

`\__ctex_ltj_pickup_alt_font:Nn` #1 是 font.id, #2 是字体名称。在 `\ltj@pickup@altfont@copy` 之中使用。

```

1024     newluacmd("__ctex_ltj_pickup_alt_font:Nn", function ()
1025         local num = scan_int()
1026         local base = scan_arg()
1027         ltjfont.pickup_alt_font_b(num, base)
1028     end, "global", "protected")

```

`\__ctex_ltj_if_alt_set:nT` 判断是否存在替代字体。会设置变量 `aftl_base` 和返回 `\@firstofone` 或 `\@gobble`。

```

1029     newluacmd("__ctex_ltj_if_alt_set:nT", ltjfont.does_alt_set, "global")

```

`\ctex_ltj_zero_globaldefs:` 设置 `\globaldefs` 为 0, 避免全局设置, 应当在分组中使用。这里通过 Lua 设置, 可以不受外部 TeX 环境中的 `\globaldefs` 的影响。

```

1030     newluacmd("ctex_ltj_zero_globaldefs:", function ()
1031         tex_set("globaldefs", 0)
1032     end, "global", "protected")
1033 }
1034 \group_end:

```

#### 15.3.4.4 字体切换方式

`\ctex_ltj_select_font:` `\CJK@family` 保存的是当前 CJK 实际的字体族名, 如果为空表示没有设置过字体。

```

\CJK@family
1035 \cs_new_protected:Npn \ctex_ltj_select_font:
1036 {
1037     \group_begin: \exp_args:Nnc \group_end:
1038     \cs_if_exist_use:NF { \l__ctex_ltj_current_font_tl }
1039     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
1040 }
1041 \tl_new:N \CJK@family
1042 \tl_new:N \l__ctex_ltj_current_font_tl
1043 \tl_set:Nn \l__ctex_ltj_current_font_tl
1044 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

`\__ctex_ltj_select_font_aux:` 使用 `\pickup@font` 取得字体名称前, 总需要先设置 `\font@name`。在这里将 `\f@family` 换成 CJK 字体族, 并确保编码正确。

```

1045 \cs_new_protected:Npn \__ctex_ltj_select_font_aux:
1046 {
1047     \group_begin:
1048     \tl_set_eq:NN \f@encoding \CJK@encoding
1049     \tl_set_eq:NN \f@family \CJK@family
1050     \cs_set_eq:NN \pickup@font \ctex_ltj_pickup_font:
1051     \__ctex_ltj_push_fontname:n { \curr@fontshape / \f@size }
1052     \ctex_ltj_pickup_font:
1053     \group_end:
1054     \font@name
1055     \__ctex_ltj_pop_fontname:

```

当字形未定义的时候, NFSS 就会启动替换机制 (`\wrong@fontshape`)。第一次启动后, `\l__ctex_ltj_current_font_tl` 还是没有定义。为此, 我们再次选择字体, 确保它有定义和指向正确的 `font.id`。这对 `AlternateFont` 的设置特别重要。

```

1056 \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
1057   { \__ctex_ltj_select_font_aux: }
1058 }
1059 \cs_new_protected:Npn \__ctex_ltj_push_fontname:n #1
1060 {
1061   \seq_gpush:No \g__ctex_ltj_fontname_seq { \font@name }
1062   \tl_gset:Ne \font@name { \exp_not:c {#1} }
1063 }
1064 \cs_new_protected:Npn \__ctex_ltj_pop_fontname:
1065 {
1066   \seq_gpop:NNT \g__ctex_ltj_fontname_seq \l__ctex_ltj_tmp_tl
1067   { \tl_gset_eq:NN \font@name \l__ctex_ltj_tmp_tl }
1068 }
1069 \seq_new:N \g__ctex_ltj_fontname_seq

```

`\ctex_ltj_pickup_font:` 替换 `\define@newfont` 内部调用的 `\extract@font` 和 `\do@subst@correction`。

```

1070 \cs_new_protected:Npn \ctex_ltj_pickup_font:
1071 {
1072   \exp_after:wN \cs_if_exist:NF \font@name
1073   {
1074     \group_begin:
1075     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
1076     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
1077     \define@newfont
1078     \group_end:
1079   }
1080 }
1081 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

`\ctex_ltj_extract_font:` LuaTeX-ja 的 `\globaljfont` 在 `luatexja-core` 中定义:

```

%%%%\jfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (false, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\protected\def\globaljfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (true, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\newluafunction\ltj@@jfont@inner
\directlua{
  local t = lua.get_functions_table()
  t[\the\ltj@@jfont@inner] = luatexja.jfont.jfontdefY
}
\def\ltj@@jfont{\luafunction\ltj@@jfont@inner}

```

`jfontdefX` 函数的作用是把 `\CS` 定义为其后的字体, `jfontdefY` 的作用是更新 JFM 和记录相关字体信息。最后的工作是:

```

tex.sprint(cat_lp, global_flag, '\protected\expandafter\def\csname ',
  (cstemp==' ') and '\space' or cstemp, '\endcsname{\ltj@cur'..
  (jfm_dir == 'yoko' and 'j' or 't') .. 'fnt', fn, '\relax}')

```

`\CS` 的作用就是把 `\ltj@curjfont` 设置为刚才定义的字体的 `font.id`。

```

1082 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
1083 {
1084   \get@external@font
1085   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
1086   {

```

```

1087     \tl_set:Nc \external@font
1088     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
1089   }
1090   \exp_after:wN \globaljfont \font@name \external@font \scan_stop:

```

这里 `\font@name` 不会直接改变当前字体,而 `\DeclareFontFamily` 和 `\DeclareFontShape` 的最后一个参数通常要使用 `\font` 来引用当前字体。为此,我们在分组内启用之前定义的字体,以便能得到正确的 `\font`。对字体参数的赋值总是全局的,不会受到分组的影响。

```

1091   \font@name
1092   \ctex_ltj_use_jfont:
1093   \use:c { \f@encoding + \f@family }
1094   \use:c { \curr@fontshape }
1095 }

```

`\ctex_ltj_use_jfont:` 使用 `jfont`, 确保当前的 `\font` 是 `jfont`。

```

1096 \cs_new_protected_nopar:Npn \ctex_ltj_use_jfont:
1097 { \tex_setfontid:D \ltj@curjfont }

```

`\ctex_ltj_subst_font:` `\do@subst@correction` 在设置通过 `sub` 或者 `ssub` 函数定义的字体时会用到。如果没有设置 `SlantedFont`, `fontspec` 会设置 `\itdefault` 作为 `\sldefault` 的替代字形,因而会用到这个函数。它的本来定义是:

```

\def\do@subst@correction{%
  \xdef\subst@correction{%
    \font@name
    \global\expandafter\font
    \csname \curr@fontshape/\f@size\endcsname
    \noexpand\fontname\font
    \relax}%
  \aftergroup\subst@correction
}

```

我们在这里不需要定义新字体,而是设置对应字体的命令。

```

1098 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
1099 {
1100   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
1101   {
1102     \group_begin:
1103     \tl_set_eq:NN \CJK@family \f@family
1104     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
1105     {
1106       \cs_gset_protected_nopar:Npx \subst@correction
1107       {
1108         \cs_new_eq:NN
1109         \exp_not:c { \l__ctex_ltj_current_font_tl }
1110         \font@name
1111       }
1112       \group_insert_after:N \group_insert_after:N
1113       \group_insert_after:N \subst@correction
1114     }
1115     \group_end:
1116   }
1117 }

```

`\ctex_ltj_if_alternate_shape_exist:nTF` 即 LuaTeX-ja 中的 `\ltj@does@alt@set`, 判断是否存在替代字体。

```

1118 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
1119 {
1120   \__ctex_ltj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
1121   \prg_return_false:
1122 }

```

```

\__ctex_ltj_patch_external_font:w 1123 \cs_new:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
1124 { \ctex_ltj_patch_external_font:n {#1} ~ at }

```

\ctex\_ltj\_select\_alternate\_font: 在 \selectfont 中更新替代字体。

```

1125 \cs_new_protected:Npn \ctex_ltj_select_alternate_font:
1126 {
1127   \ctex_ltj_if_alternate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
1128   {
1129     \ctex_ltj_pickup_alt_font:nn
1130     { \l__ctex_ltj_current_shape_tl } { \f@size }
1131   }
1132 }
1133 \tl_new:N \l__ctex_ltj_current_shape_tl
1134 \tl_set:Nn \l__ctex_ltj_current_shape_tl
1135 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

\ltj@pickup@altfont@auxy 被用在函数 output\_alt\_font\_cmd 中,作用是定义替代字体。

```

1136 \cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
1137 {
1138   \cs_if_exist:cF { #1/\f@size }
1139   {
1140     \group_begin:
1141     \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
1142     \__ctex_ltj_push_fontname:n { \curr@fontshape / \f@size }
1143     \ctex_ltj_pickup_font:
1144     \group_end:
1145     \__ctex_ltj_pop_fontname:
1146   }
1147 }

```

\ltj@pickup@altfont@copy 被用在函数 pickup\_alt\_font\_a 中。 \ltj@@getjfontnumber 的作用是将字体命令 #1 对应的 font.id 保存到 \ltj@tempcntc 中。

```

1148 \cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2
1149 {
1150   \ltj@@getjfontnumber #1
1151   \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
1152 }

```

### 15.3.4.5 数学字体族

以下内容来自 lltjfont.sty,目的是让汉字可以在数学环境中直接使用。

\ctex\_ltj\_if\_jfont:nTF 参数 #1 是一个 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 编码名称或者字体命令。L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 字体命令的一般形式是:

$$\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle$$

通过截取名字中的  $\langle encoding \rangle$  来判断是否是 jfont。最后会设置 \ifin@ 为对应的 \iftrue 或者 \iffalse。

```

1153 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
1154 {
1155   \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
1156   \legacy_if:nTF { in@ }
1157 }
1158 \cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
1159 {#1}

```

\ctex\_ltj\_if\_jfont\_math:NTF #1 是一个形式为 M@ $\langle encoding \rangle$  的命令,它由 \DeclareFontEncoding 的第三个参数来定义。

```

1160 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1

```

```

1161 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
1162 \group_begin:
1163   \char_set_catcode_other:N M
1164   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
1165     { \ctex_ltj_if_jfont:nTF {#3} }
1166 \group_end:

```

`\getanddefine@fonts` 在使用的场合, `\escapechar` 已经被设置成 `-1`, 使用 `\token_to_str:N` 就可以得到名字, 不必使用 `\cs_to_str:N`。

```

1167 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
1168 {
1169   \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
1170     { \ctex_ltj_get_and_define_fonts_ja:nN }
1171     { \ctex_ltj_get_and_define_fonts_al:nN }
1172     {#1} #2
1173 }
1174 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
1175 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
1176 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
1177 {
1178   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
1179   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
1180   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
1181   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
1182   \tl_gset:Ne \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
1183   \ctex_ltj_pickup_font:
1184   \tl_put_right:Ne \math@fonts
1185     {
1186       \ltj@setpar@global
1187       \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
1188       \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
1189       \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
1190     }
1191 }

```

`\use@mathgroup` 在使用 `unicode-math` 宏包时, `\ctex_ltj_math_group_hook:` 将被重定义。

```

1192 \cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
1193 {
1194   \mode_if_math:T
1195   {
1196     \math@bgroup
1197     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
1198     \ctex_ltj_math_group_hook:
1199     \ctex_ltj_if_jfont_math:NTF #1
1200       { \jfam } { \mathgroup } #2 \scan_stop:
1201     \math@egroup
1202   }
1203 }
1204 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
1205 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

对 `unicode-math` 的补丁主要是将 `unicode-math-table.tex` 中的数学符号设置为 `lua-texja` 中的数学字母。本段代码应放在 `\ctex_ltj_math_group_hook:` 的定义之后, 避免因宏包载入顺序而造成的编译错误。

```

1206 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
1207 {
1208   \group_begin:
1209     \cs_set_protected:Npn #1 ##1##2##3
1210       { \ltjsetmathletter { ##1 } }
1211     #2
1212   \group_end:
1213 }

```

```

1214 \ctex_at_end_package:nn { unicode-math }
1215 {
1216   \cs_if_exist:NTF \um_input_math_symbol_table:
1217     {
1218       \ctex_ltj_set_math_letter:NN
1219       \um_sym:nnn
1220       \um_input_math_symbol_table:
1221     }
1222     {
1223       \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
1224       \cs_set_protected:Npn \ctex_ltj_math_group_hook:
1225         { \__um_switchto_literal: }
1226       \ctex_ltj_set_math_letter:NN
1227       \__um_sym:nnn
1228       \__um_input_math_symbol_table:
1229     }
1230 }

```

### 15.3.4.6 字体族的定义与使用

`\ctex_set_jfm:n` 设置 JFM, 需要进行一些重定向操作。

```

\l__ctex_ltj_jfm_tl
\l__ctex_ltj_redirect_jfm_prop
1231 \cs_new_protected:Npn \ctex_set_jfm:n #1
1232 {
1233   \prop_get:NnNF \l__ctex_ltj_redirect_jfm_prop {#1} \l__ctex_ltj_jfm_tl
1234   { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
1235 }
1236 \cs_generate_variant:Nn \ctex_set_jfm:n { o }
1237 \prop_new:N \l__ctex_ltj_redirect_jfm_prop
1238 \prop_set_from_keyval:Nn \l__ctex_ltj_redirect_jfm_prop
1239 {
1240   plain      = mono ,
1241   quanjiao   = zh_CN / quanjiao ,
1242   banjiao    = zh_CN / banjiao ,
1243   kaiming    = zh_CN / kaiming
1244 }
1245 \keys_define:nn { ctex_ltj / fontspec }
1246 {
1247   JFM .code:n = \ctex_set_jfm:n {#1} ,
1248   JFM .value_required:n = true
1249 }
1250 \tl_new:N \l__ctex_ltj_jfm_tl

```

注意这里的`@@=ctex_ltj`。

```

1251 \ctex_set_jfm:o { \l__ctex_punct_tl }

```

`\CJK@encoding` 在  $\text{\LaTeX}$  下,  $\text{\LuaTeX-j}$ a 依赖字体编码来实现特殊设置。例如上述的 `\ctex_ltj_if_jfont:nTF` 就是通过判断编码来实现的, 它在设置数学字体时会用到。所以不应该与西文共用 EU2。定义字体族 `song` 为 `\CJK@encoding` 的默认替换字体。下划线 `_` 不在 `\nfss@catcodes` 里, 可以放心使用。

```

1252 \tl_const:Nn \CJK@encoding { LTJY3 }
1253 \DeclareFontEncoding { \CJK@encoding } { } { }
1254 \use:e
1255 {
1256   \exp_not:N \DeclareFontSubstitution
1257     { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1258 }
1259 \ctex_ltj_add_kyenc:n { \CJK@encoding }
1260 \DeclareFontFamily { \CJK@encoding } { song } { }
1261 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1262   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1263 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }

```

```

1264 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1265 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
1266 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
1267   { \CJK@encoding } { song } { \mdefault } { \shapedefault }
1268 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
1269   { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1270 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
1271 \jfam \c__ctex_ltj_math_fam_int

```

这是 `luatexja-fontspec` 中新增的一些字体选项。

```

1272 \newfontfeature { CID } { cid = #1 }
1273 \newfontfeature { JFM } { jfm = #1 }
1274 \newfontfeature { JFM-var } { jfmvar = #1 }

```

在新版本的 `fontspec` 中, `\__fontspec_fontname_wrap:n` 变成了私有函数。

```

1275 \keys_define:nn { fontspec-preparse-external }
1276 {
1277   NoEmbed .code:n =
1278     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
1279 }
1280 \cs_new:Npn \__ctex_ltj_noembed_wrap:n #1 { psft: #1 }

```

`\ctex_ltj_set_family:nnn` 将自定义的字体族名与 `fontspec` 实际设置的名字对应起来。

```

1281 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1282 {
1283   \group_begin:
1284   \clist_clear:N \l__ctex_ltj_char_range_clist
1285   \prop_clear:N \l__ctex_ltj_alternate_prop
1286   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1287   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1288   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1289   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1290   {
1291     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1292     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1293       {#1} { \l__ctex_ltj_font_options_clist }
1294     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1295     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1296     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1297   }
1298   \group_end:
1299 }
1300 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1301 \clist_new:N \l__ctex_ltj_font_options_clist

```

`\__ctex_ltj_use_global_options:N` 应用默认字体选项, 并总是设置 JFM 和 NFSSEncoding。

```

1302 \cs_new_protected:Npn \__ctex_ltj_use_global_options:N #1
1303 {
1304   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1305   \clist_put_left:Ne #1
1306     { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
1307 }

```

`\g__ctex_ltj_family_name_prop` 分别保存 `fontspec` 设置的字体族名、字体名称和字体选项。

```

\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop
1308 \prop_new:N \g__ctex_ltj_family_name_prop
1309 \prop_new:N \g__ctex_ltj_family_font_name_prop
1310 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

`\__ctex_ltj_check_family:n` 删除重复的定义, 清除替代字体的先前设置。

```

1311 \cs_new_protected:Npn \__ctex_ltj_check_family:n #1

```

```

1312 {
1313   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1314   {
1315     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1316     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1317     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1318     {
1319       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1320       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1321       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1322       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1323     }
1324     \msg_warning:nnee { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1325   }
1326 }
1327 \tl_new:N \l__ctex_ltj_tmp_tl
1328 \msg_new:nnn { ctex } { redefine-family }
1329 { Redefining`CJKfamily`~\__ctex_ltj_msg_family_map:n {#1}'~(#2). }

```

`\__ctex_ltj_gset_family_cs:nn` 在设置字体时,实际上并不是马上就定义。而是只保存相关参数,在通过 `\CJKfamily` 第一次使用时才定义。需要注意将编码改为 `\CJK@encoding`。

```

1330 \cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1331 {
1332   \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
1333   {
1334     \group_begin:
1335     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1336     \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1337     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1338     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1339     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1340     \__ctex_ltj_set_alternate_family:n {#1}
1341     \group_end:
1342   }
1343 }
1344 \tl_new:N \l__ctex_ltj_base_family_tl
1345 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1346 \cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1347 \cs_new_protected:Npn \__ctex_ltj_set_alternate_family:n #1
1348 {
1349   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1350   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1351   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1352   \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1353 }
1354 \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

`\CJKfamily` 切换字体。

```

1355 \NewDocumentCommand \CJKfamily { m }
1356 { \ctex_ltj_switch_family:e {#1} \tex_ignorespaces:D }
1357 \cs_new_protected:Npn \ctex_ltj_switch_family:n #1
1358 {
1359   \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1360   {
1361     \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1362     \selectfont
1363   }
1364   { \__ctex_ltj_family_unknown_warning:n {#1} }
1365 }
1366 \tl_new:N \l__ctex_ltj_family_tl
1367 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { e }

```

`\ctex_ltj_family_if_exist:nNTF` 判断 CJK 字体族#1 是否存在,若存在则把实际族名保存到#2 中。

```

1368 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }

```

```

1369 {
1370   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1371   { \prg_return_true: }
1372   {
1373     \cs_if_exist_use:CTF { \__ctex_ltj_family_csname:n {#1} }
1374     {
1375       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1376       \prg_return_true:
1377     }
1378     { \prg_return_false: }
1379   }
1380 }
1381 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { e } { T , F , TF }

```

```

\__ctex_ltj_family_unknown_warning:n 1382 \cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
1383 {
1384   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1385   {
1386     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1387     {
1388       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1389       \msg_warning:nnn { ctex } { family-unknown } {#1}
1390     }
1391   }
1392 }
1393 \seq_new:N \g__ctex_ltj_unknown_family_seq
1394 \msg_new:nnn { ctex } { family-unknown }
1395 {
1396   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1397   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1398 }
1399 \cs_new:Npn \__ctex_ltj_msg_def_family_map:n #1
1400 {
1401   \str_case_e:nnF {#1}
1402   {
1403     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1404     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1405     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1406   }
1407   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1408   [...] \{ ... \}
1409 }
1410 \cs_new:Npn \__ctex_ltj_msg_family_map:n #1
1411 {
1412   \str_case_e:nnF {#1}
1413   {
1414     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1415     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1416     \CJKttdefault { \token_to_str:N \CJKttdefault }
1417   }
1418   {#1}
1419 }

```

```

\ctex_ltj_fontspec:nn 1420 \cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
1421 {
1422   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1423   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1424   { \ctex_ltj_switch_family:e { \l_ctex_ltj_family_tl } }
1425   {
1426     \int_gincr:N \g__ctex_ltj_family_int
1427     \__ctex_ltj_fontspec:enn
1428     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1429     {#1} {#2}
1430   }
1431 }
1432 \cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
1433 { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }

```

```

1434 \cs_new_protected:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1435 {
1436   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1437   {
1438     \cs_if_free:cF
1439     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1440     {
1441       \cs_gset_eq:cc
1442       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1443       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1444       \cs_gset_eq:cc
1445       { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1446       { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1447     }
1448     \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1449   }
1450   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1451   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1452   \ctex_ltj_switch_family:n {#1}
1453 }
1454 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { e }
1455 \prop_new:N \g__ctex_ltj_fontspec_prop

```

```

\ctex_ltj_add_font_features:n 1456 \cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1457 { \ctex_ltj_add_font_features:en { \l_ctex_ltj_family_tl } {#1} }
1458 \cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
1459 {
1460   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1461   {#1} \l__ctex_ltj_tmp_tl
1462   {
1463     \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1464     {#1} \l__ctex_ltj_font_options_clist
1465     \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1466     \bool_set_true:N \l__ctex_ltj_add_alternate_bool
1467     \ctex_ltj_fontspec:ee
1468     { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1469     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1470   }
1471   { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1472 }
1473 \bool_new:N \l__ctex_ltj_add_alternate_bool
1474 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { e }
1475 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { e }
1476 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1477 {
1478   \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1479   It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1480 }

```

\\_\_ctex\_ltj\_pass\_args:nnnn 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, 自带展开功能。

```

1481 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1482 {
1483   \tl_if_novalue:nTF {#2}
1484   { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1485   {
1486     \use:e { #1 {#2} {#3} }
1487     #4
1488   }
1489 }
1490 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1491 {
1492   \use:e { #1 {#4} {#2} }
1493   #3
1494 }

```

```

\setCJKfamilyfont 1495 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures

```

```

1496 {
1497   \__ctex_ltj_pass_args:nnnn
1498   { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1499   { }
1500 }
1501 \msg_new:nnn { ctex } { command-already-defined }
1502 { Control~sequence~#1~is~already~defined. }
1503 \NewDocumentCommand \newCJKfontfamily { o m o m }
1504 {
1505   \tl_set:Ne \l__ctex_ltj_tmp_tl
1506   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
1507   \cs_if_exist:NTF #2
1508   { \msg_error:nne { ctex } { command-already-defined }
1509     { \token_to_str:N #2 } }
1510   {
1511     \cs_set_protected:Npx #2
1512     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1513   }
1514   \__ctex_ltj_pass_args:nnnn
1515   { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1516   { }
1517 }
1518 \NewDocumentCommand \CJKfontspec { o m }
1519 {
1520   \__ctex_ltj_pass_args:nnnn
1521   { \ctex_ltj_fontspec:nn } {#1} {#2}
1522   { \tex_ignorespaces:D }
1523 }
1524 \NewDocumentCommand \addCJKfontfeatures { m }
1525 {
1526   \ctex_ltj_add_font_features:e {#1}
1527   \tex_ignorespaces:D
1528 }
1529 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1530 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1531 {
\setCJKmonofont 1532   \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1533   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1534   { \normalfont }
1535 }
1536 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1537 \NewDocumentCommand \setCJKsansfont { o m }
1538 {
1539   \__ctex_ltj_pass_args:nnnn
1540   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1541   { \normalfont }
1542 }
1543 \NewDocumentCommand \setCJKmonofont { o m }
1544 {
1545   \__ctex_ltj_pass_args:nnnn
1546   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1547   { \normalfont }
1548 }
1549 \NewDocumentCommand \setCJKmathfont { o m }
1550 {
1551   \__ctex_ltj_pass_args:nnnn
1552   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1553   { }
1554 }
1555 \NewDocumentCommand \defaultCJKfontfeatures { m }
1556 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1557 \clist_new:N \g__ctex_ltj_default_features_clist
1558 \@onlypreamble \setCJKmainfont
1559 \@onlypreamble \setCJKsansfont
1560 \@onlypreamble \setCJKmonofont
1561 \@onlypreamble \setCJKmathfont

```

```

1562 \@onlypreamble \setCJKromanfont
1563 \@onlypreamble \defaultCJKfontfeatures

```

\ctex\_ltj\_ensure\_default\_family: 在导言区结束确认 \CJKfamilydefault 确实存在。

```

1564 \cs_new_protected:Npn \ctex_ltj_ensure_default_family:
1565   {
1566     \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1567     {
1568       \ctex_ltj_family_if_exist:eNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1569       {
1570         \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmddefault }
1571         { \use:n }
1572         {
1573           \ctex_ltj_family_if_exist:eNTF { \CJKrmddefault } \l__ctex_ltj_tmp_tl
1574           { \tl_gset:Nn \CJKfamilydefault { \CJKrmddefault } \use_none:n }
1575           { \use:n }
1576         }
1577         {
1578           \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1579           {
1580             \prop_map_break:n
1581             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1582           }
1583         }
1584       }
1585       \normalfont
1586       \ctex_ltj_update_mathfont:
1587     }
1588   }

```

\ctex\_ltj\_update\_mathfont: 更新数学字体为实际的字体。

```

1589 \cs_new_protected:Npn \ctex_ltj_update_mathfont:
1590   {
1591     \ctex_ltj_family_if_exist:eNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1592     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1593     {
1594       \ctex_ltj_family_if_exist:eNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1595       { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1596     }
1597   }
1598 \cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
1599   {
1600     \tl_const:Ne \c__ctex_ltj_math_family_tl {#1}
1601     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1602     { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1603     \cs_if_free:cTF
1604     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
1605     {
1606       \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1607       { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1608     }
1609     {
1610       \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1611       { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
1612     }
1613   }

```

### 15.3.4.7 替代字体的设置

**AlternateFont** 设置替代字体的选项。  
**CharRange**

```

1614 \keys_define:nn { ctex_ltj / fontspec }
1615   {

```

```

1616 AlternateFont .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
1617 AlternateFont .value_required:n = true ,
1618 CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1619 CharRange .value_required:n = true
1620 }

```

\ctex\_ltj\_set\_alternate\_prop:n 保存替代字体序列。

```

1621 \cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
1622 { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
1623 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
1624 {
1625   \clist_set:Ne \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
1626   \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
1627   \exp_args:No \__ctex_ltj_push_alternate_prop:nn
1628     { \l__ctex_ltj_tmp_clist } {#1}
1629 }
1630 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
1631 {
1632   \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
1633   \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
1634 }
1635 \clist_new:N \l__ctex_ltj_tmp_clist
1636 \prop_new:N \l__ctex_ltj_alternate_prop

```

\ctex\_ltj\_set\_alternate\_family:nnF 如果在字体的选项中设置了 CharRange, 则只设置替代字体。

```

1637 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnF
1638 {
1639   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1640     { \__ctex_ltj_set_family_aux:nnn }
1641     { \__ctex_ltj_set_alternate_family_aux:nnn }
1642 }
1643 \cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
1644 {
1645   \__ctex_ltj_check_family:n {#1}
1646   \prop_if_empty:NF \l__ctex_ltj_alternate_prop
1647     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1648   #3
1649 }
1650 \cs_new_protected:Npn \__ctex_ltj_set_alternate_family_aux:nnn #1#2#3
1651 { \ctex_ltj_set_alternate_family:nn {#1} {#2} }

```

\ctex\_ltj\_save\_alternate\_seq:Nn 保存由 AlternateFont 设置的替代字体序列。

```

1652 \cs_new_protected:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1653 {
1654   \prop_map_inline:Nn \l__ctex_ltj_alternate_prop
1655     { \__ctex_ltj_save_alternate_auxi:w ##2 { } \q_mark #1 {#2} }
1656 }
1657 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxi:w #1#2#
1658 {
1659   \tl_if_blank:nTF {#2}
1660     { \__ctex_ltj_save_alternate_auxii:w {#1} }
1661     { \__ctex_ltj_save_alternate_auxii:w {#1} {#2} }
1662 }
1663 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxii:w #1#2#3 #4 \q_mark #5#6
1664 {
1665   \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
1666   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#3}
1667   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1668   \tl_if_blank:nTF {#2}
1669     { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
1670     {
1671       \tl_set:Ne \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
1672       \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}

```

```

1673     }
1674     \use:e
1675     {
1676         \ctex_ltj_save_altername_family:Nnnn \exp_not:N #5
1677         { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1678         { \exp_not:o { \l__ctex_ltj_altername_options_clist } }
1679         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1680     }
1681 }
1682 \clist_new:N \l__ctex_ltj_altername_options_clist
1683 \cs_generate_variant:Nn \ctex_ltj_save_altername_seq:Nn { c }

```

`\ctex_ltj_set_altername_family:nn` 设置选项 CharRange 范围内的替代字体。如果已经定义了主字体，我们也马上定义替代字体，否则只保存起来备用。

```

1684 \cs_new_protected:Npn \ctex_ltj_set_altername_family:nn #1#2
1685 {
1686     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1687     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1688     \ctex_ltj_set_altername_family:coonn
1689     { \__ctex_ltj_altername_cs:n {#1} }
1690     { \l__ctex_ltj_char_range_clist }
1691     { \l__ctex_ltj_font_options_clist } {#2} {#1}
1692 }
1693 \cs_new_protected:Npn \ctex_ltj_set_altername_family:Nnnnn #1#2#3#4#5
1694 {
1695     \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1696     { \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} }
1697     \ctex_ltj_save_altername_family:Nnnn #1 {#2} {#3} {#4}
1698 }
1699 \cs_generate_variant:Nn \ctex_ltj_set_altername_family:Nnnnn { coo }

```

`\ctex_ltj_save_altername_family:Nnnn` 保存替代字体序列的定义，以备定义主字体时使用。

```

1700 \cs_new_protected:Npn \ctex_ltj_save_altername_family:Nnnn #1#2#3#4
1701 {
1702     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1703     \cs_gset_protected:Npx #1
1704     { \exp_not:o { #1 \ctex_ltj_set_altername_family:nnn {#2} {#3} {#4} } }
1705 }

```

`\ctex_ltj_set_altername_family:nnn` 实际定义替代字体族。

```

1706 \cs_new_protected:Npn \ctex_ltj_set_altername_family:nnn #1#2#3
1707 {
1708     \group_begin:
1709     \cs_set_eq:NN \CJKfamily \use_none:n
1710     \ctex_ltj_swap_cs:NN
1711     \DeclareFontShape@ \ctex_ltj_declare_altername_shape:nnnnnn
1712     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1713     \fontspec_set_family:Nnn \l__ctex_ltj_altername_family_tl {#2} {#3}
1714     \group_end:
1715 }
1716 \tl_new:N \l__ctex_ltj_altername_family_tl

```

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1717 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1718 {
1719     \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1720     \cs_set_eq:NN #1 #2
1721     \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1722     \cs_undefine:N \__ctex_ltj_tmp:w
1723 }

```

LTJFONTUID `fontspec` 在一个字体族的选项和字体名称相同的时候,就不定义新字体。为了避免混淆替代字体的设置,我们新定义一个虚拟的选项 `LTJFONTUID`, 确保 `fontspec` 对 CJK 字体族总是定义新字体。

```

1724 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1725 \cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
1726   {
1727     \int_gincr:N \g__ctex_ltj_family_int
1728     \clist_put_right:Ne #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1729   }
1730 \int_new:N \g__ctex_ltj_family_int

```

在定义替代字体的字形时,通过字符范围与主字体的对应字形关联起来。`\DeclareFontShape@` 一个有六个参数,我们只需要使用它的第三个参数(`series`) 和第四个参数(`shape`)。

```

1731 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1732   {
1733     \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1734     \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1735     { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
1736   }

```

与 `LuaTeX-ja` 的 `\DeclareAlternateKanjiFont` 的功能类似,区别是固定编码为 `\CJK@encoding`。这个设置总是全局的。

```

1737 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1738   {
1739     \clist_map_inline:Nn #1
1740     {
1741       \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1742       {
1743         \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1744         \l__ctex_ltj_char_range_tl
1745       }
1746       { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1747     }
1748     \__ctex_ltj_save_alternate_shape:cn
1749     { \__ctex_ltj_alternate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1750     { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
1751   }

```

我们使用 `->` 而不是像 `LuaTeX-ja` 一样使用 `-` 作为区间的分隔符。`LuaTeX-ja` 支持使用负数来引用由 `JFM` 设置的字符类。如果使用 `-` 作为分隔符,那么负数单独使用时,就需要把它放在两层花括号之内(例如`{-1}`),或者使用类似`{-1}{-1}` 的形式才不会解释错误。

```

1752 \NewDocumentCommand \ctex_ltj_set_alternate_shape:nnn
1753   { m m > { \SplitArgument { 1 } { -> } } m }
1754   { \ctex_ltj_set_alternate_shape:nnnn {#1} {#2} #3 }
1755 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nnnn #1#2#3#4
1756   {
1757     \ctex_ltj_set_alternate_shape:e
1758     {
1759       \__ctex_ltj_range_normalization:nn {#3} {#4}
1760       { \CJK@encoding / \exp_not:n {#2} }
1761       { \CJK@encoding / \exp_not:n {#1} }
1762     }
1763   }
1764 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:n #1
1765   {
1766     \ctex_ltj_set_alt_font:nnnn #1
1767     \__ctex_ltj_save_alternate_shape:cn
1768     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1769     { \ctex_ltj_set_alt_font:nnnn #1 }
1770   }
1771 \cs_generate_variant:Nn \ctex_ltj_set_alternate_shape:n { e }

```

`\ctex_ltj_set_alternate_shape:nN` 若字符范围预先由 `declarecharrange` 声明, 则可以直接使用。

```

1772 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:nN #1#2#3
1773 {
1774   \tl_map_inline:Nn #3
1775   {
1776     \ctex_ltj_set_alternate_shape:n
1777     {
1778       ##1
1779       { \CJK@encoding/#2 }
1780       { \CJK@encoding/#1 }
1781     }
1782   }
1783 }

```

`\__ctex_ltj_save_alternate_shape:Nn` 将实际设置的替换字形保存起来用于清除或恢复。暂时令 `\l__ctex_ltj_base_family_tl` 为 `\scan_stop:` 是让它不被展开, 使得替换字体的设置可以在 `\addCJKfontfeature` 中直接使用。

```

1784 \cs_new_protected:Npn \__ctex_ltj_save_alternate_shape:Nn #1#2
1785 {
1786   \group_begin:
1787   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1788   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1789   \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }
1790   \group_end:
1791 }
1792 \cs_generate_variant:Nn \__ctex_ltj_save_alternate_shape:Nn { c }

```

`clearalternatefont` 清除和重置操作总是全局的。  
`resetalternatefont`

```

1793 \ctex_define:n
1794 {
1795   clearalternatefont .code:n =
1796   { \clist_map_function:eN {#1} \ctex_ltj_clear_alternate_font:n } ,
1797   resetalternatefont .code:n =
1798   { \clist_map_function:eN {#1} \ctex_ltj_reset_alternate_font:n } ,
1799   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1800   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1801 }
1802 \cs_new_protected:Npn \ctex_ltj_clear_alternate_font:n #1
1803 {
1804   \group_begin:
1805   \ctex_ltj_family_if_exist:eNTF {#1} \l__ctex_ltj_base_family_tl
1806   {
1807     \cs_if_exist_use:cT
1808     { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1809     {
1810       \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1811       {#1} { \l__ctex_ltj_base_family_tl }
1812       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1813       \selectfont
1814     }
1815   }
1816   { \__ctex_ltj_family_unknown_warning:n {#1} }
1817   \group_end:
1818 }
1819 \cs_new_protected:Npn \ctex_ltj_reset_alternate_font:n #1
1820 {
1821   \group_begin:
1822   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1823   {
1824     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1825     \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1826     \selectfont
1827   }

```

```

1828   \group_end:
1829 }
1830 \prop_new:N \g__ctex_ltj_reset_alterate_prop
1831 \cs_generate_variant:Nn \clist_map_function:nN { e }

```

`declarecharrange` 预先声明字符范围。

```

1832 \ctex_define:n
1833 {
1834   declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
1835   declarecharrange .value_required:n = true
1836 }
1837 \cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
1838 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1839 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
1840 \cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
1841 { \tl_trim_spaces_apply:nN {#1} \ctex_ltj_declare_char_range:nn }

```

`\ctex_ltj_declare_char_range:nn` #1 是名字, #2 是范围。

`\g__ctex_ltj_char_range_prop`

```

1842 \cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
1843 {
1844   \tl_clear:N \l__ctex_ltj_char_range_tl
1845   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1846   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1847   \ctex_ltj_def_char_range_key:n {#1}
1848   \tl_clear:N \l__ctex_ltj_char_range_tl
1849 }
1850 \tl_new:N \l__ctex_ltj_char_range_tl
1851 \prop_new:N \g__ctex_ltj_char_range_prop

```

`\ctex_ltj_save_char_range:n` 预先解释字符区间的意义。

```

1852 \NewDocumentCommand \ctex_ltj_save_char_range:n
1853 { > { \SplitArgument { 1 } { -> } } m }
1854 { \ctex_ltj_save_char_range:nn #1 }
1855 \cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
1856 {
1857   \tl_put_right:Ne \l__ctex_ltj_char_range_tl
1858   { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
1859 }
1860 \cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
1861 {
1862   \tl_if_novalue:nTF {#2}
1863   {
1864     { \int_eval:n {#1} }
1865     { \int_eval:n {#1} }
1866   }
1867   {
1868     { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
1869     { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
1870   }
1871 }
1872 \int_const:Nn \c__ctex_ltj_range_min_int { "80 }
1873 \int_const:Nn \c__ctex_ltj_range_max_int { \c_max_char_int }

```

`\ctex_ltj_def_char_range_key:n` 在字体设置选项中定义字符范围键。

```

1874 \cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
1875 {
1876   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1877   {
1878     \keys_define:nn { ctex_ltj / fontspec }
1879     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1880   }
1881 }

```

`\ctex_ltj_char_range_key:nn` 如果字符范围键没有值, 则只设置的这个字符范围内的替代字体。

```

1882 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1883 {
1884   \tl_if_blank:nTF {#2}
1885     { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1886     { \__ctex_ltj_push_alternate_prop:nn {#1} { {#1} #2 } }
1887 }

```

### 15.3.4.8 其他设置

在抄录环境中禁用 `autospacing` 和 `autoxspacing`。然而, LuaTeX-ja 还是会使 JAchar 自动折行。没有看到有简单的禁用折行的办法, 可能需要设置所有的 JAchar 的 `prebreakpenalty` 或 `postbreakpenalty` 为 10000:

```

\directlua
{
  luatexja.isglobal = tex.globaldefs > 0 and "global" or ""
  for i = 0x80, 0x10FFFF do
    if luatexja.charrange.jcr_table_main[i] > 0 and
       luatexja.charrange.jcr_table_main[i] < 218 and
       luatexja.charrange.is_japanese_char_curlist(i) then
      luatexja.stack.set_stack_table(luatexja.stack_table_index.PRE + i, 10000)
    end
  end
}

1888 \AtBeginDocument
1889 {
1890   \ctex_appto_cmd:NnnTF \verbatim@font
1891     { \char_set_catcode_letter:n { 64 } }
1892     { \CTEX@verbatim@font@hook }
1893     { }
1894     { \ctex_patch_failure:N \verbatim@font }
1895 }
1896 \cs_new_protected:Npn \CTEX@verbatim@font@hook
1897 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

```

LuaTeX-ja 的 `lltjcore.sty` 会将 L<sup>A</sup>T<sub>E</sub>X 的 `\verb` 中 `\null` (即 `\hbox{}`) 替换为 `\vadjust{}`, 因为空 `\hbox{}` 会阻断 `xkanjiskip` 的自动插入。由于 `ctex` 禁用了 `ltj-latex`, 这里需要自行复制该修复。

```

1898 \@namedef { ver@lltjcore.sty } { }
1899 \if@compatibility \else
1900   \def \verb
1901     {
1902       \relax \ifmmode \hbox \else \leavevmode \vadjust { } \fi
1903       \bgroup
1904         \verb@eol@error \let \do \@makeother \dospecials
1905         \verbatim@font \@noligs
1906         \language \l@nohyphenation
1907         \@ifstar \@sverb \@verb
1908     }
1909 \fi
1910 \patchcmd { \do@noligs } { \kern \z@ } { \vadjust { } } { } { }

```

`\@@italiccorr` L<sup>A</sup>T<sub>E</sub>X 的倾斜校正也要重新定义。

```

1911 \cs_set_eq:NN \@@italiccorr \/

```

`\ctex_ltj_set_kanjiskip:N` `\ltjsetkanjiskip` 和 `\ltjsetxkanjiskip` 是相应的 `\ltjsetparameter` 的快捷方式, 在使用他们时, 要注意先使用 `\ltj@setpar@global`。

```

1912 \cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N

```

```

1913 { \ltj@setpar@global \ltjsetkanjiskip }
1914 \cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
1915 { \ltj@setpar@global \ltjsetxkanjiskip }

1916 <@@-ctex>
1917 </luatex>

```

### 15.3.5 ctex-engine-uptex.def

```
1918 <*uptex|aptex>
```

按 CJK 的命名习惯模拟 \CJKfamily。

```

1919 \NewDocumentCommand \CJKfamily { m }
1920 { \kanjifamily {#1} \selectfont }

```

将up $\LaTeX$  的默认字体由mc 改为zhrm, 并启用 \jfam。

```

1921 \DeclareErrorKanjiFont {JY2}{zhrm}{m}{n}{10}
1922 \DeclareKanjiSubstitution {JY2}{zhrm}{m}{n}
1923 \DeclareKanjiSubstitution {JT2}{zhrm}{m}{n}
1924 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1925 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1926 \jfam \symmincho

```

\em 取消up $\LaTeX$  对 \em 使用\mcfamily、\gtfamily 命令的重定义, 恢复 $\LaTeX 2_{\epsilon}$  对 \em 的原始定义。如果用户已经重定义了 \em, 则新定义保持不变。up $\LaTeX$  2016/05/07u00 的定义有所变化, 这一行为可以由用户通过 platexrelease 包改变, 需要分支处理。

```

1927 \ctex_patch_cmd_once:NnnTF \em
1928 { \ExplSyntaxOff }
1929 { \emminershape \else \gtfamily \itshape }
1930 { \emminershape \else \itshape }
1931 { }
1932 {
1933   \ctex_patch_cmd:Nnn \em
1934     { \mcfamily \upshape \else \gtfamily \itshape }
1935     { \emminershape \else \itshape }
1936 }
1937 \cs_set_nopar:Npn \emminershape { \upshape }

```

\ctex\_set\_upfamily:nnn 将 NFSS 字体族#1 设置为 JFM 字体名#2, 粗体形式字体名#3。其中字体名形如upzhserif, 不包括表示方向的后缀-h 与-v。粗体字体名为空时不设置该字形。本命令不设置字体映射, 需要复用已有的字体映射或另行设置。

```

1938 \cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
1939 {
1940   \DeclareKanjiFamily{JY2}{#1}{}
1941   \DeclareKanjiFamily{JT2}{#1}{}
1942   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1943   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1944   \tl_if_empty:nF {#3}
1945   {
1946     \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{}
1947     \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{}
1948     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{}
1949     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{}
1950   }
1951 }

```

\ctex\_set\_upmap:nnn 设置up $\TeX$  字体映射。#1 是形如upserif 的 PS TFM 字体名, 不带表示粗体的后缀b 与表示排版方向的后缀-h 与-v。#2 与#3 是普通与粗体的实际字体名。

```
1952 \cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
```

```

1953 {
1954   \ctex_set_zhmap:n
1955   {
1956     \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
1957     \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
1958     \tl_if_empty:nF {#3}
1959     {
1960       \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
1961       \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
1962     }
1963   }
1964 }

```

`\ctex_set_upmap_unicode:nnn` 设置 $\upTeX$  字体映射, 使用`unicode CMap`。参数同上。

```

1965 \cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
1966 {
1967   \ctex_set_zhmap:n
1968   {
1969     \special { pdf:mapline~#1-h~unicode~#2 }
1970     \special { pdf:mapline~#1-v~unicode~#2 }
1971     \tl_if_empty:nF {#3}
1972     {
1973       \special { pdf:mapline~#1b-h~unicode~#3 }
1974       \special { pdf:mapline~#1b-v~unicode~#3 }
1975     }
1976   }
1977 }

```

`\ctex_set_upfonts:nnnnn` 设置 $\upTeX$  基本字体映射, 按 `zhmetrics-uptex` 的定义, 依次设置衬线体正、粗、意大利, 无衬线体正、粗, 等宽体正——共 6 种字体, 并分横排及直排。

```

1978 \cs_new_protected:Npn \ctex_set_upfonts:nnnnn #1#2#3#4#5#6
1979 {
1980   \ctex_set_upmap:nnn { upserif } {#1} {#2}
1981   \ctex_set_upmap:nnn { upserifit } {#3} {}
1982   \ctex_set_upmap:nnn { upsans } {#4} {#5}
1983   \ctex_set_upmap:nnn { upmono } {#6} {}
1984 }

```

以下命令只能在导言区使用。

```

1985 \@onlypreamble \ctex_set_upfamily:nnn
1986 \@onlypreamble \ctex_set_upmap:nnn
1987 \@onlypreamble \ctex_set_upmap_unicode:nnn
1988 \@onlypreamble \ctex_set_upfonts:nnnnn

```

`everyysel` 宏包(2011/10/28)未考虑 $\up\LaTeX$  对 `\selectfont` 的修改, 需要引入 `pxeveryysel` 宏包。

```

1989 \bool_if:NT \c__ctex_everyysel_loaded_bool
1990 { \RequirePackage { pxeveryysel } }

1991 <uptex|aptex>

```

### 15.3.6 修改主要字体命令

修改 `\rmfamily` 等主要字体命令, 使得中文字体能随西文主要字体更新。使用  $\LaTeX$  2020/10/01 提供的 `rmfamily` 等 NFSS 钩子。

`xeCJK` 和 `zhmCJK` 已经有相同的工作, 本段代码不需要对他们使用。

```

1992 <*pdfTeX|luatex|uptex|aptex>
<pdfTeX> 1993 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool

```

`\ctex_provide_font_hook:NNN` 给 `\rmfamily` 等字体命令加钩子, 钩子名字统一为 `\CTEX@rmfamilyhook` 等。  
`\CTEX@rmfamilyhook`

```

1994 \cs_new_protected:Npn \ctex_provide_font_hook:NNN #1#2
1995 {
1996   \exp_args:Nc \__ctex_provide_font_hook_aux:NNNN
1997   { CTEX \cs_to_str:N #2 } #1#2
1998 }
1999 \cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
2000 {
2001   \tl_new:N #1
2002   \exp_args:Ne \ctex_gadd_ltxhook:nn { \cs_to_str:N #2 } {#1}
2003 }
2004 \ctex_provide_font_hook:NNN \rmfamily \@rmfamilyhook \selectfont
2005 \ctex_provide_font_hook:NNN \sffamily \@sffamilyhook \selectfont
2006 \ctex_provide_font_hook:NNN \ttfamily \@ttfamilyhook \selectfont
⟨pdfTeX|luatex⟩ 2007 \ctex_provide_font_hook:NNN \normalfont \@defaultfamilyhook \usefont

```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```

2008 \tl_if_exist:NF \CJKfamilydefault
2009 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
2010 ⟨*pdfTeX|luatex⟩
2011 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
2012 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
2013 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
2014 \tl_gput_right:Nn \CTEX@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
2015 \tl_gput_right:Nn \CTEX@sffamilyhook { \CJKfamily { \CJKsfdefault } }
2016 \tl_gput_right:Nn \CTEX@ttfamilyhook { \CJKfamily { \CJKttdefault } }
2017 \tl_gput_right:Nn \CTEX@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }
2018 ⟨pdfTeX|luatex⟩

```

up<sup>L</sup>TeX 不需要补丁 `\normalfont`, 只需要修改 `\kanjifamilydefault`。

```

2019 ⟨*uptex|aptex⟩
2020 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
2021 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
2022 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhht } }
2023 \tl_gput_right:Nn \CTEX@rmfamilyhook { \kanjifamily { \CJKrmdefault } }
2024 \tl_gput_right:Nn \CTEX@sffamilyhook { \kanjifamily { \CJKsfdefault } }
2025 \tl_gput_right:Nn \CTEX@ttfamilyhook { \kanjifamily { \CJKttdefault } }
2026 \tl_gset:Nn \kanjifamilydefault { \CJKfamilydefault }
2027 ⟨uptex|aptex⟩

```

zhmCJK 判断结束。

⟨pdfTeX⟩ 2028 \fi:

使修改立刻生效, 保证导言区字体族正确。

```
2029 \normalfont
```

在导言区末尾更新 `\CJKfamilydefault`, pdfTeX 已经在之前使用过此处代码。

⟨pdfTeX⟩ 2030 \ctex\_at\_end\_preamble:n { \ctex\_update\_default\_family: }

`\ctex_update_default_family:` 在导言区结束, 如果 `\CJKfamilydefault` 没有被更改, 则在此时根据西文字体的情况更新 `\CJKfamilydefault`。xeCJK 已经有这个功能, 不需要再调整。

```

2031 \cs_new_protected:Npn \ctex_update_default_family:
2032 {
2033   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
2034   {
2035     \group_begin:
2036     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
2037     \tl_gset:Ne \CJKfamilydefault

```

```

2038         {
2039             \str_case:onF { \familydefault }
2040             {
2041                 { \rmdefault } { \exp_not:N \CJKrmdefault }
2042                 { \sfdefault } { \exp_not:N \CJKsfdefault }
2043                 { \ttdefault } { \exp_not:N \CJKttdefault }
2044             }
2045             { \CJKfamilydefault }
2046         }
2047     \group_end:
2048 }

```

使用 Lua $\LaTeX$  时, 自动调整得到的 `\CJKfamilydefault` 可能没有定义, 需要确认它的存在性。使用 `CJK` 宏包或 `up $\LaTeX$`  时, `C19rm`、`JY2rm` 等总是有定义的, 不需要确认。

```

<luatex> 2049 \ctex_ltj_ensure_default_family:
2050 }

```

`\l__ctex_family_default_init_tl` 往 `\CJKfamilydefault` 中加入标志, 用于判断它是否被更改。

```

2051 \tl_new:N \l__ctex_family_default_init_tl
2052 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
2053 \tl_set:Ne \l__ctex_family_default_init_tl
2054 {
2055     \exp_not:N \__ctex_family_default_wrap:n
2056     { \exp_not:o { \CJKfamilydefault } }
2057 }
2058 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
2059 </pdfTeX|luatex|uptex|aptex>

```

### 15.3.7 hyperref 兼容性处理

现在处理各个引擎下的 PDF 中文书签问题。根据编译引擎与文件编码的不同, `ctex` 向 `hyperref` 传递适当的参数, 完成中文书签的正确设置。用户仍需要自己载入 `hyperref` 宏包。

`\ctex_hypersetup:n` 如果已经载入 `hyperref` 宏包, 则直接使用其定义设置选项; 否则 `\ctex_hypersetup:n` 的效果与 `\PassOptionsToPackage` 一致, 只传递宏包参数。如果用户不载入 `hyperref` 宏包, 相关参数即被丢弃。

```

2060 <*class|ctex>
2061 \@ifpackageloaded { hyperref }
2062 {
2063     \cs_new_protected:Npn \ctex_hypersetup:n #1
2064     { \hypersetup {#1} }
2065 }
2066 {
2067     \cs_new_protected:Npn \ctex_hypersetup:n #1
2068     { \PassOptionsToPackage {#1} { hyperref } }
2069 }
2070 </class|ctex>

```

在 `pdf $\TeX$`  下使用 GBK 编码, `DVIPDFMx` 驱动可以直接用它的 `\special` 命令, 其他模式用 `xCJK2uni` 宏包处理。使用 UTF-8 编码时, `CJKutf8` 已经处理了书签问题, 但仍需要设置 `pdfencoding` 为 `unicode`, 目的是在书签的开头写入 BOM (`\376\377`), 提示这是 UTF-16BE 字节流。`hyperref` 2021-02-04 版开始默认设置 `unicode` 为 `true`, 对于 `DVIPDFMx` 驱动, 我们需要禁用这个设置, 为此设置 `pdfencoding` 为 `pdfdoc`。

`driverfallback` 是 `hyperref` 的加载选项, 加载后即被禁用, 不能通过 `\hypersetup` 设置。因此当 `hyperref` 已被其他宏包(如 `beamer`)加载时, 只能跳过此选项。

```

2071 <*pdfTeX>

```

```

2072 \ifpackageloaded { hyperref }
2073   { }
2074   { \PassOptionsToPackage { driverfallback = dvipdfmx } { hyperref } }
2075 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
2076   {
2077     \ctex_hypersetup:n { CJKbookmarks = true }
2078     \sys_if_output_pdf:TF
2079       { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
2080       {
2081         \ctex_hypersetup:n { pdfencoding = pdfdoc }
2082         \ctex_at_end_package:nn { hyperref }
2083         {
2084           \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
2085             {
2086               \ctex_at_shipout_first:n
2087                 { \special { pdf:tounicode~GBK-EUC-UCS2 } }
2088             }
2089           { \RequirePackage { xCJK2uni } }
2090         }
2091       }
2092   }
2093   { \ctex_hypersetup:n { pdfencoding = unicode } }
2094 </pdfTEX>

```

X<sub>q</sub>TeX 和 LuaTeX 统一设置 pdfencoding 为 unicode。

```

2095 <*xetex|luatex>
2096 \ctex_hypersetup:n { pdfencoding = unicode }
2097 </xetex|luatex>

```

我们假定 upTeX 使用 DVIPDFMx 驱动输出，于是使用与 pdfTeX 类似的设置。注意 upTeX 需要使用 UTF8-UTF16 的编码转换。

```

2098 <*uptex|aptex>
2099 \ifpackageloaded { hyperref }
2100   { \ctex_hypersetup:n { pdfencoding = pdfdoc } }
2101   {
2102     \ctex_hypersetup:n
2103       {
2104         driverfallback = dvipdfmx ,
2105         pdfencoding     = pdfdoc
2106       }
2107   }
2108 \ctex_at_end_package:nn { hyperref }
2109   {
2110     \ctex_at_shipout_first:n
2111       { \special { pdf:tounicode~UTF8-UTF16 } }
2112   }
2113 </uptex|aptex>
2114 <*pdfTEX|xetex|luatex|uptex|aptex>

```

### 15.3.8 CJKfntef、xeCJKfntef 相关设置

对 pdfTeX 与 X<sub>q</sub>TeX 引擎，分别在 CJKfntef、xeCJKfntef 宏包的末尾关闭彩色显式等多余格式。

```

2115 <*pdfTEX>
2116 \ctex_at_end_package:nn { CJKfntef }
2117   {
2118     \normalem
2119     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
2120       { \tl_clear:c { CJK#1color } }
2121 </pdfTEX>
2122 <*xetex>

```

```

2123 \ctex_at_end_package:nn { xeCJKfntef }
2124 {
2125   \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
2126   {
2127     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
2128     { \xeCJKsetup { #1 / format = { } } }
2129   }
2130   {
2131     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
2132     { \tl_clear:c { CJK#1color } }
2133   }
2134 </xetex>
2135 <*pdfTeX|xetex>
2136   \clist_map_inline:nn
2137   { underdot , underline , underdblline , underwave , sout , xout }
2138   { \__ctex_clear_fntef_color:n {#1} }
2139 }
2140 </pdfTeX|xetex>

```

### 15.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 2141 \cs_new_protected:Npn \ctex_update_ccwd:
\ccwd 2142 <*pdfTeX|xetex>
2143 {
2144   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
2145   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
2146 }
2147 </pdfTeX|xetex>
2148 <*luatex>
2149 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
2150 </luatex>
2151 <*uptex|aptex>
2152 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
2153 </uptex|aptex>
2154 \dim_new:N \ccwd

```

\ctex\_update\_ccglue: 更新字间距。

```

2155 \cs_new_protected:Npn \ctex_update_ccglue:
2156 <*pdfTeX|xetex>
2157 {
2158   \cs_set_protected:Npn \CJKglue
2159   { \skip_horizontal:N \l__ctex_ccglue_skip }
2160   \cs_if_exist:NT \__ctex_ccglue:
2161   { \cs_set_eq:NN \__ctex_ccglue: \CJKglue }
2162 }
2163 </pdfTeX|xetex>
2164 <*luatex>
2165 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
2166 </luatex>
2167 <*uptex|aptex>
2168 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
2169 </uptex|aptex>
2170 \skip_new:N \l__ctex_ccglue_skip

```

\ctex\_if\_ccglue\_touched\_p: 检查用户是否修改过汉字间距。

\ctex\_if\_ccglue\_touched: TF

```

2171 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
2172 {
2173 <*pdfTeX|xetex>
2174   \if_meaning:w \CJKglue \__ctex_ccglue:
2175   \prg_return_false: \else: \prg_return_true: \fi:
2176 </pdfTeX|xetex>
2177 <*luatex>
2178   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }

```

```

2179     { \prg_return_false: } { \prg_return_true: }
2180 </luatex>
2181 <*uptex|aptex>
2182     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
2183     { \prg_return_false: } { \prg_return_true: }
2184 </uptex|aptex>
2185 }
<pdf|xetex> 2186 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
2187 <*pdf|xetex>
2188 \ctex_at_end:n
2189 {
2190     \cs_set_protected:Npn \__ctex_update_stretch_auxii:
2191     {
2192         \ctex_if_ccglue_touched:TF
2193         { \ctex_update_ccwd: }
2194         { \__ctex_update_stretch_auxiii: }
2195     }
2196 }
2197 </pdf|xetex>

```

`\ctex_update_em_unit:` 将当前汉字的宽度保存到 `\ccwd` 中备用。不采用 `1em`, 因为这时的 `1em` 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 `.tfm` 字体上表现更明显。在 `pdfTeX` 和 `XYTeX` 下, 直接使用 `\f@size\p@` 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 `XYTeX` 可以用 `\fontcharwd` 来改进。而在 `pdfTeX` 下, 若使用 `zhmetrics` 技术, 所有的汉字共享同一个 `.tfm`, `\fontcharwd` 也就没有意义。在 `LuaTeX` 下, `LuaTeX-ja` 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 `\zw` 作为汉字宽度。`upTeX` 可以直接使用原生的长度单位 `zw`。

```

2198 \cs_new_protected:Npn \ctex_update_em_unit:
<pdf|xetex> 2199 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 2200 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 2201 { \dim_set:Nn \ccwd { 1zw } }

```

### 15.3.10 其他

```

\ctex_add_to_selectfont:n 2202 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
\CTEX@selectfont@hook 2203 {
2204     \cs_set_protected:Npx \CTEX@selectfont@hook
2205     { \exp_not:o { \CTEX@selectfont@hook #1 } }
2206 }
2207 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:

```

使用 `everyysel` 包的情况。`\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 `LuaTeX-ja` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。`everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。`pxeveryysel` 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。`KOMA-Script` 宏包也会进行检查, 我们需要小心处理。

```

2208 \bool_if:NTF \c__ctex_everyysel_loaded_bool
2209 {
2210     \cs_if_free:NF \@EverySelectfont@Init
2211     {
2212         \group_begin:
2213         \cs_set:Npn \__ctex_tmp:N #1
2214         {
2215             \tl_set:Nn \l__ctex_tmp_tl {#1}
2216             \cs_new_eq:NN \CTEX@selectfont@save #1
2217             \cs_new_protected:Npn \__ctex_restore_selectfont:
2218             {
2219                 \cs_if_free:NF \scr@new@selectfont
2220                 {

```

CJK 直接修改 `\selectfont` 和 `pxeveryysel` 的补丁, 会使 KOMA-Script 的 `\par@update` 失效。

```

2221 <*pdfTeX|uTeX|apTeX>
2222             \cs_if_free:NF \par@update
2223 <*uTeX|apTeX>
2224             {
2225                 \tl_put_right:Nn \@EverySelectfont@Init
2226                 { \tl_put_right:Nn #1 { \par@update } }
2227             }
2228 </uTeX|apTeX>
2229 <*pdfTeX>
2230             { \tl_put_right:Nn #1 { \par@update } }
2231             \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
2232 </pdfTeX>
2233 </pdfTeX|uTeX|apTeX>
2234             \cs_set_eq:NN \scr@new@selectfont #1
<!pdfTeX> 2235             \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
2236             }
2237             \tl_put_left:Nn \@EverySelectfont@Init
2238             { \cs_set_eq:NN #1 \CTEX@selectfont@save }
2239             \cs_undefine:N \__ctex_restore_selectfont:
2240         }
2241     }
2242     \ctex_parse_name:NN \__ctex_tmp:N \selectfont
2243     \exp_last_unbraced:NNo \group_end:
2244     \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
2245     { \ExplSyntaxOff }
2246     { \size@update }
2247     { \CTEX@selectfont@hook \size@update }
2248     { \__ctex_restore_selectfont: }
2249     { \ctex_patch_failure:N \selectfont }
2250 }
2251 \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
2252 { \EverySelectfont }
2253 }

```

使用 L<sup>A</sup>T<sub>E</sub>X 2021-06-01 的新钩子, 不使用 `everyysel` 包的情况。

```

2254 {
2255     \cs_new_protected:Npn \ctex_gadd_selectfont_hook:n
2256     { \ctex_gadd_ltxhook:nn { selectfont } }
2257 }

```

`\CJK@plane` 有定义, 说明处于 CJK 宏包的 `\CJKsymbol` 之内, 不必使用钩子。

```

2258 <*pdfTeX>
2259 \ctex_gadd_selectfont_hook:n
2260 { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2261 </pdfTeX>
2262 <*xetex|luatex|uTeX|apTeX>
2263 \ctex_gadd_selectfont_hook:n { \CTEX@selectfont@hook }
2264 </xetex|luatex|uTeX|apTeX>

```

Attribute 寄存器 `\ltj@curjfont` 的初始值是 `-1`, 必须把它设置为一个有效的 `font.id`, 否则编译时会直接退出。

```

2265 <*luatex>
2266 \ctex_add_to_selectfont:n
2267 {
2268     \ctex_ltj_select_font:
2269     \ctex_ltj_select_alternate_font:
2270 }
2271 \tl_set:Nn \CJK@family { song } \selectfont
2272 \tl_clear:N \CJK@family
2273 </luatex>

```

`\ctex_update_xkanjiskip:` `upTeX` 和 `LuaTeX-ja` 对 `\xkanjiskip` 都是即时赋值。单位 `zw` 与字体相关, 因此需要每次 `\l__ctex_xkanjiskip_skip` `\selectfont` 的时候更新一次 `\xkanjiskip`。如果用户设置过 `\xkanjiskip`, 就不更新。注意,

同 TeX 的 `\baselineskip` 一样,如果在一个段落内多次设置了 `\kanjiskip` 或 `\xkanjiskip`,只有最后的设置会影响全段。

```

2274 <*luatex|uptex|aptex>
2275 \cs_new_protected:Npn \ctex_update_xkanjiskip:
2276 {
2277   \skip_if_eq:nnT
<luatex> 2278     { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
<uptex|aptex> 2279     { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2280   {
2281     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
<luatex> 2282     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
<uptex|aptex> 2283     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2284   }
2285 }
2286 \tl_new:N \l__ctex_xkanjiskip_tl
2287 \tl_set:Nn \l__ctex_xkanjiskip_tl
<luatex> 2288   { .25\zw plus 1pt minus 1pt }
<uptex|aptex> 2289   { .25zw plus 1pt minus 1pt }
2290 \skip_new:N \l__ctex_xkanjiskip_skip
2291 \skip_set:Nn \l__ctex_xkanjiskip_skip
<luatex> 2292   { \ltjgetparameter { xkanjiskip } }
<uptex|aptex> 2293   { \tex_xkanjiskip:D }

2294 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
2295 </luatex|uptex|aptex>

```

`\cht` 分别从 `.jfm` 中读取字符高度、深度和宽度,目前仅考虑横排的情况。

```

\cdp
\c wd
\ctex_update_kanjisize:
2296 <*luatex>
2297 \dim_new:N \cht
2298 \dim_new:N \cdp
2299 \dim_new:N \c wd
2300 \group_begin:
2301 \char_set_catcode_space:n { 32 }
2302 \lua_now:e
2303 {
2304   local nulltable = { }
2305   local fmt = luatexja.jfont.font_metric_table
2306   local getattribute = tex.getattribute
2307   local setdimen = tex.setdimen
2308   ctex.newluacmd("ctex_update_kanjisize:", function ()
2309     local ft = fmt[getattribute("ltj@curjfont")] or nulltable
2310     local ft = ft and ft.char_type or nulltable
2311     local fk = ft and ft[0] or nulltable
2312     setdimen("cht", fk.height or 0)
2313     setdimen("cdp", fk.depth or 0)
2314     setdimen("c wd", fk.width or ft.zw or 0)
2315     end, "global", "protected")
2316   }
2317 \group_end:
2318 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2319 </luatex>

```

`space` 在导言区或正文中设置忽略空格方式。pdfTeX 和 XeTeX 下初始设置为 `auto`, LuaTeX、upTeX 下是无效选项。

```

2320 \ctex_define:n
2321 {
2322 <*pdf|etex|etex>
2323   space .choice: ,
2324   space / true .code:n =
<pdf|etex> 2325     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<etex> 2326     { \xeCJKsetup { CJKspace = true } } ,
2327   space / auto .code:n =
<pdf|etex> 2328     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,

```

```

<xetex> 2329     { \xeCJKsetup { CJKspace = false } } ,
2330     space / false .code:n =
<pdfTeX> 2331     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xetex> 2332     { \xeCJKsetup { CJKspace = false } } ,
2333     space .default:n = { true } ,
2334     space .initial:n = { auto }
2335 </pdfTeX|xetex>
2336 <*luatex|uptex|aptex>
2337     space .code:n =
2338     { \msg_warning:nn { ctex } { invalid-option } }
2339 </luatex|uptex|aptex>
2340   }

```

**punct** 在导言区或正文中设置标点符号输出格式。LuaTeX-ja 设置的是字体的默认 JFM, 只会影响到之后设置的字体。upTeX 暂时无效。

```

2341 \ctex_define:n
2342 {
2343   punct .code:n =
2344   {
2345     \tl_set:Nc \l__ctex_punct_tl {#1}
<pdfTeX> 2346     \punctstyle { \l__ctex_punct_tl }
<xetex> 2347     \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
<luatex> 2348     \ctex_set_jfm:o { \l__ctex_punct_tl }
<uptex|aptex> 2349     \msg_warning:nn { ctex } { invalid-option }
2350   } ,
2351   punct .default:n = { quanjiao } ,
2352 }

```

**experiment** 实验性选项。experiment/CJKecglue 用于统一设置 CJK 文字与西文之间的间距。XeTeX 下转  
**experiment/CJKecglue** 发给 xeCJK, LuaTeX 和 upTeX 下设置 xkanjiskip, pdfTeX 下不支持。

```

2353 \ctex_define:n
2354 { experiment .meta:nn = { ctex / experiment } {#1} }
2355 \keys_define:nn { ctex / experiment }
2356 {
2357 <*xetex>
2358   CJKecglue .code:n =
2359   { \xeCJKsetup { CJKecglue = { \skip_horizontal:n {#1} } } } ,
2360 </xetex>
2361 <*luatex|uptex|aptex>
2362   CJKecglue .code:n =
2363   {
2364     \tl_set:Nc \l__ctex_xkanjiskip_tl {#1}
2365     \skip_set:Nc \l__ctex_xkanjiskip_skip {#1}
<luatex> 2366     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
<uptex|aptex> 2367     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
2368   } ,
2369 </luatex|uptex|aptex>
2370 <*pdfTeX>
2371   CJKecglue .code:n =
2372   { \msg_warning:nn { ctex } { CJKecglue-unsupported } } ,
2373 </pdfTeX>
2374   CJKecglue .value_required:n = true
2375 }

2376 </pdfTeX|xetex|luatex|uptex|aptex>

```

### 15.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

<class|ctex> 2377 \ctex_file_input:n { \c__ctex_engine_file_str }

```

## 15.4 用户设置接口

```
\ctexset 2378 <*class|ctex|ctexheading>
2379 \NewDocumentCommand \ctexset { } { \ctex_set:n }
2380 </class|ctex|ctexheading>
```

`\CTEXsetup` 过时命令。  
`\CTEXoptions`

```
2381 <*class|ctex>
2382 \NewDocumentCommand \CTEXsetup { +0 { } } > { \TrimSpaces } m }
2383 {
2384   \tl_if_blank:nTF {#1}
2385     { \ctex_deprecated_command:Nn \CTEXsetup { } }
2386     {
2387       \ctex_deprecated_command:Nn \CTEXsetup
2388         { \ctexset {~#2~::~{~#1~}}~is~set. }
2389       \ctex_set:nn {#2} {#1}
2390     }
2391 }
2392 \NewDocumentCommand \CTEXoptions { +0 { } }
2393 {
2394   \tl_if_blank:nTF {#1}
2395     { \ctex_deprecated_command:Nn \CTEXoptions { } }
2396     {
2397       \ctex_deprecated_command:Nn \CTEXoptions
2398         { \ctexset {~#1~}~is~set. }
2399       \ctex_set:n {#1}
2400     }
2401 }
```

## 15.5 字距与缩进

`autoindent` `autoindent` 也是可以用在正文中的选项,意义与宏包选项`option/autoindent` 相同。

```
2402 \ctex_define:n
2403 {
2404   autoindent .choice: ,
2405   autoindent .default:n = { true } ,
2406   autoindent / true .code:n =
2407     {
2408       \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2409       \ctex_select_size:
2410     } ,
2411   autoindent / false .code:n =
2412     { \tl_clear:N \l__ctex_autoindent_tl } ,
2413   autoindent / unknown .code:n =
2414     {
2415       \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2416       \ctex_select_size:
2417     }
2418 }
```

`\CTEXsetfont` 无论字体大小是否变化都更新相关信息。

```
2419 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2420 \cs_new_protected:Npn \ctex_select_size:
2421 { \cs_if_free:NNTF \size@update { \ctex_update_size: } { \selectfont } }
```

`\ctex_update_size:` 在字号变化时更新 `\ccwd`、`\parindent` 和汉字间距。字距为零则恢复正常设置。

```
2422 \cs_new_protected:Npn \ctex_update_size:
2423 {
2424   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2425   {
```

```

2426     \ctex_update_stretch:
2427     \ctex_update_parindent:
2428     }
2429     { \ctex_update_ziju: }
2430   }
2431 \tl_const:Nc \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2432 \tl_new:N \l__ctex_ziju_tl
2433 \tl_set_eq:NN \l__ctex_ziju_tl \c__ctex_zero_tl

```

在 `\selectfont` 中, 若 `\size@update` 为 `\relax`, 说明字体大小没有变化, 我们也就不用更新相关参数。

```

2434 \ctex_add_to_selectfont:n
2435 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

**linestretch** 若行宽不是汉字宽度的整数倍, 自然要求伸展它们之间的差。这里设置的是在此基础上的额外伸展量。初始化为一个汉字的宽度。若设置为 `\maxdimen`, 则禁用此功能。参数的默认单位是汉字的宽度 `\ccwd`。

```

2436 \ctex_define:n
2437 {
2438   linestretch .code:n =
2439   {
2440     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2441     \ctex_select_size:
2442   },
2443   linestretch .value_required:n = true
2444 }
2445 \tl_new:N \l__ctex_line_stretch_tl
2446 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

`\ctex_update_stretch:` 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 $\epsilon$ -TeX 的除法是四舍五入, 而我们这里应该用截断。由于没有可展性的要求, 直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`, 则只更新 `\ccwd`, 否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前, `\baselineskip` 还未更新, 不能直接使用它。

```

2447 \cs_new_protected:Npn \ctex_update_stretch:
2448 {
2449   \ctex_update_em_unit:
2450   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2451   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2452     { \__ctex_update_stretch_auxi: }
2453     { \__ctex_update_stretch_auxii: }
2454 }
2455 \cs_new_protected:Npn \__ctex_update_stretch_auxi:
2456 {
2457   \ctex_if_ccglue_touched:TF
2458     { \ctex_update_ccwd: }
2459     {
2460       \dim_set:Nn \l__ctex_tmp_dim
2461         { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2462       \skip_set:Nn \l__ctex_ccglue_skip
2463         { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2464       \ctex_update_ccglue:
2465     }
2466 }
2467 \cs_new_protected:Npn \__ctex_update_stretch_auxii:
2468 { \__ctex_update_stretch_auxiii: }
2469 \cs_new_protected:Npn \__ctex_update_stretch_auxiii:
2470 {
2471   \int_set:Nn \l__ctex_tmp_int

```

```

2472     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2473     \tex_divide:D \l__ctex_tmp_int \ccwd
2474     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2475     {
2476         \skip_set:Nn \l__ctex_ccglue_skip
2477         {
2478             \c_zero_dim plus \dim_eval:n
2479             {
2480                 ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2481                 \l__ctex_tmp_int
2482             }
2483         }
2484     }
2485     { \skip_zero:N \l__ctex_ccglue_skip }
2486     \ctex_update_ccglue:
2487 }

```

`\ctex_update_parindent:` 更新段落首行缩进。此函数在字号变化时调用。

```

2488 \cs_new_protected:Npn \ctex_update_parindent:
2489 {
2490     \tl_if_empty:NF \l__ctex_autoindent_tl
2491     {
2492         \dim_compare:nNnF \parindent = \c_zero_dim
2493         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2494     }
2495 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2496 \NewDocumentCommand \ziju { m }
2497 { \exp_args:Ne \ctex_ziju:n {#1} \tex_ignorespaces:D }
2498 \cs_new_protected:Npn \ctex_ziju:n #1
2499 {
2500     \tl_set:Ne \l__ctex_ziju_tl { \fp_eval:n {#1} }
2501     \ctex_select_size:
2502 }

```

`\ctex_update_ziju:` 更新字距。若字距不大于  $-1$ , 即 `\ccwd` 为非正值, 则不计算伸缩值。否则, 首先假定汉字的宽度为正常宽度加上字距, 看一行上能正常放下多少个汉字。

```

2503 \cs_new_protected:Npn \ctex_update_ziju:
2504 {
2505     \ctex_update_em_unit:
2506     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2507     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2508     \dim_compare:nNnTF \ccwd > \c_zero_dim

```

伸展量保证行内的剩余空白能够被均匀地填充到汉字之间, 收缩的最大限度是让当前行还能够再挤下一个汉字并且不会出现负间距。由  $\text{T}_{\text{E}}\text{X}$  决定伸展还是收缩。

```

2509     {
2510         \dim_set:Nn \l__ctex_tmp_dim
2511         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2512         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2513         \tex_divide:D \l__ctex_tmp_int \ccwd
2514         \dim_sub:Nn \l__ctex_tmp_dim { \l__ctex_tmp_int \ccwd }

```

由于 `\parindent` 是一个固定值, 并不参与伸缩, 容易导致第一行出现坏盒子。我们在这里将字数减去 2, 以此放大伸缩值。

```

2515     \dim_compare:nNnF \parindent = \c_zero_dim
2516     {
2517         \int_compare:nNnF \l__ctex_tmp_int < 3
2518         { \int_sub:Nn \l__ctex_tmp_int { 2 } }

```

```

2519     }
2520     \skip_set:Nn \l__ctex_ccglue_skip
2521     {
2522         \l__ctex_ziju_dim
2523         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2524         minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2525         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2526     }
2527 }
2528 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2529 \ctex_update_ccglue:

```

字距设置得比较大时,为了尽量保证段首缩进能够与下一行对齐,应该需要相应地加上或者减去伸缩值。但是这里并不清楚 TeX 是伸展还是收缩,之前以“当前行是否还放得下一个汉字”为标准加上或减去伸缩值的做法也未必与实际结果一致,所以只好还是设置为 $2\ccwd$ 。

```

2530 \ctex_update_parindent:
2531 }
2532 \dim_new:N \l__ctex_ziju_dim

```

`\CTEXindent` 过时命令。  
`\CTEXnoindent`

```

2533 \NewDocumentCommand \CTEXindent { }
2534 {
2535     \ctex_deprecated_command:Nn \CTEXindent
2536     { \parindent is~set~to~2\ccwd. }
2537     \ctex_update_ccwd:
2538     \dim_set:Nn \parindent { 2 \ccwd }
2539 }
2540 \NewDocumentCommand \CTEXnoindent { }
2541 {
2542     \ctex_deprecated_command:Nn \CTEXnoindent
2543     { \parindent is~set~to~0pt. }
2544     \dim_zero:N \parindent
2545 }

```

## 15.6 中文数字与日期

需要注意的是, `ltkeys` 设置的选项列表是 `\@raw@opt@(<\@currname>).\@currentx)`, 该列表不会将 `\PassOptionsToPackage` 传递的选项完全展开。

```

2546 \exp_args:Ne \PassOptionsToPackage
2547 { encoding = \g__ctex_encoding_tl }
2548 { zhnumber }
2549 \RequirePackage { zhnumber }

```

```

\chinese 2550 \cs_new:Npn \chinese { \zhnum_counter:n }
2551 \cs_new_eq:NN \@chinese \@zhnum
2552 \cs_new_eq:NN \Chinese \chinese
2553 \cs_new_eq:NN \CTEXcounter \use_none:n

```

给 `enumitem` 宏包注册 `\chinese`、`\zhnum` 和 `\zhdig`。

```

2554 \ctex_at_end_package:nn { enumitem }
2555 {
2556     \cs_if_free:NF \AddEnumerateCounter
2557     {
2558         \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2559         \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2560         \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2561     }
2562 }

```

```

\CTEXnumber 2563 \NewDocumentCommand \CTEXnumber { m m }
\CTEXdigits 2564 { \protected@edef #1 { \zhnumber {#2} } }
                2565 \NewDocumentCommand \CTEXdigits { m m }
                2566 { \protected@edef #1 { \zhdigits {#2} } }

today 2567 \cs_set_eq:NN \CTEX@todayold \today
        2568 \ctex_define:n
        2569 {
        2570     today .choice: ,
        2571     today / old .code:n =
        2572         { \cs_set_eq:NN \today \CTEX@todayold } ,
        2573     today / small .code:n =
        2574         {
        2575             \cs_set_eq:NN \today \zhtoday
        2576             \zhnumsetup { time = Arabic }
        2577         } ,
        2578     today / big .code:n =
        2579         {
        2580             \cs_set_eq:NN \today \zhtoday
        2581             \zhnumsetup { time = Chinese }
        2582         } ,
        2583     today / unknown .code:n =
        2584         { \msg_error:nne { ctex } { today-undef } {#1} }
        2585     }
        2586 \msg_new:nnnn { ctex } { today-undef }
        2587 { Today~format~`#1'~is~undefined. }
        2588 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

## 15.7 其他中文标题定义

`\proofname` `\proofname` 未在标准文档类中定义, 需要确保它非空。

```

2589 \tl_if_exist:NF \proofname
2590 {
2591     \tl_new:N \proofname
2592     \tl_set:Nn \proofname { Proof }
2593 }

2594 \ctex_define:n
2595 {
2596     contentsname .tl_set:N = \contentsname ,
2597     listfigurename .tl_set:N = \listfigurename ,
2598     listtablename .tl_set:N = \listtablename ,
2599     figurename .tl_set:N = \figurename ,
2600     tablename .tl_set:N = \tablename ,
2601     abstractname .tl_set:N = \abstractname ,
2602     indexname .tl_set:N = \indexname ,
2603     appendixname .tl_set:N = \appendixname ,
2604     proofname .tl_set:N = \proofname ,
<article> 2605     bibname .tl_set:N = \refname
<book|report> 2606     bibname .tl_set:N = \bibname
2607 <*beamer>
2608     algorithmname .tl_set:N = \algorithmname ,
2609     bibname .tl_set:N = \bibname ,
2610     refname .tl_set:N = \refname ,
2611     continuation .tl_set:N = \insertcontinuationtext
2612 </beamer>
2613 }

2614 <*ctex>
2615 \msg_new:nnn { ctex } { ctexbibname }
2616 {
2617     Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2618     The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2619 }
2620 \tl_if_exist:NTF \insertcontinuationtext

```

```

2621 {
2622   \ctex_define:n
2623   {
2624     algorithmname .tl_set:N = \algorithmname ,
2625     bibname       .tl_set:N = \bibname ,
2626     refname       .tl_set:N = \refname ,
2627     continuation .tl_set:N = \insertcontinuationtext
2628   }
2629 }
2630 {
2631   \tl_if_exist:NTF \bibname
2632   { \ctex_define:n { bibname .tl_set:N = \bibname } }
2633   {
2634     \tl_if_exist:NTF \refname
2635     { \ctex_define:n { bibname .tl_set:N = \refname } }
2636     {
2637       \msg_warning:nn { ctex } { ctexbibname }
2638       \ctex_define:n { bibname .tl_set:N = \ctexbibname }
2639     }
2640   }
2641 }
2642 </ctex>

2643 </class|ctex>

```

## 15.8 中文化的标题结构

本节内容在 C<sub>T</sub>E<sub>X</sub> 文档类或打开 heading 选项下生效。

```
2644 <*class|heading>
```

### 15.8.1 定义标题格式选项

\c\_\_ctex\_section\_headings\_seq 保存 \section 级以下标题名字。

```

2645 <*article|book|report>
2646 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2647 { section , subsection , subsubsection , paragraph , subparagraph }
2648 </article|book|report>

```

\c\_\_ctex\_headings\_seq 保存各级标题名字。

```

<article|book|report|beamer> 2649 \seq_const_from_clist:Nn \c__ctex_headings_seq
2650 <*article|book|report>
2651 {
2652   part ,
<book|report> 2653   chapter ,
2654   section , subsection , subsubsection , paragraph , subparagraph
2655 }
2656 </article|book|report>
<beamer> 2657 { part , section , subsection }

```

```

\__ctex_initial_heading:n 2658 \cs_new_protected:Npn \__ctex_initial_heading:n #1
2659 {
2660   \tl_new:c { CTEX@pre#1 }
2661   \tl_new:c { CTEX@post#1 }
2662   \tl_const:ce { CTEX@the#1 }
2663   {
2664     \exp_not:c { CTEX@pre#1 }
2665     \exp_not:c { CTEX@the#1 }
2666     \exp_not:c { CTEX@post#1 }
2667   }
2668   \tl_const:ce { CTEX@#1name }

```

```

2669 {
2670   \group_begin:
2671   \exp_not:c { CTEX@#1@nameformat }
2672   {
2673     \exp_not:c { CTEX@pre#1 }
2674     \exp_not:N \tl_if_empty:NTF
2675     \exp_not:c { CTEX@#1@numberformat }
2676     { \exp_not:c { CTEX@the#1 } }
2677     {
2678       \group_begin:
2679       \exp_not:c { CTEX@#1@numberformat }
2680       \exp_not:c { CTEX@the#1 }
2681       \group_end:
2682     }
2683     \exp_not:c { CTEX@post#1 }
2684   }
2685   \group_end:
2686 }
2687 }

```

```

\__ctex_def_heading_keys:n 2688 \cs_new_protected:Npn \__ctex_def_heading_keys:n #1
2689 {
2690   \exp_args:NNe \tl_put_right:Nn \l__ctex_tmp_tl
2691   {
2692     #1 .meta:nn = { ctex / #1 } { ##1 } ,
2693     #1 / name .code:n =
2694     { \ctex_assign_heading_name:nn {#1} { ##1 } } ,
2695     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2696     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2697     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2698     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2699     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2700     #1 / numbering .initial:n = true ,
2701     #1 / beforeskip .initial:n = \c_zero_skip ,
2702     #1 / afterskip .initial:n = \c_zero_skip ,
2703     #1 / indent .initial:n = \c_zero_dim ,
2704     #1 / beforeskip .value_required:n = true ,
2705     #1 / afterskip .value_required:n = true ,
2706     #1 / indent .value_required:n = true ,
2707     <*article|book|report>
2708     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2709     #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2710     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2711     #1 / hang .initial:n = true ,
2712     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2713     #1 / tocline .cs_set:Np = \exp_not:c { CTEX@#1@tocline } ##1##2 ,
2714     \__ctex_plus_key_aux:nn {#1} { break } ,
2715     </article|book|report>
2716     \__ctex_plus_key_aux:nn {#1} { format } ,
2717     \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2718     \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2719     \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2720     \__ctex_plus_key_aux:nn {#1} { aftername } ,
2721     \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2722   }
2723 }
2724 \cs_new:Npn \__ctex_plus_key_aux:nn #1#2
2725 {
2726   #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2727   #1 / #2 + .code:n =
2728   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } } ,
2729   #1 / #2 ~ + .code:n =
2730   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { ##1 } }
2731 }

```

\ctex\_assign\_heading\_name:nn **name** 的值是一个至多两个元素的逗号分隔列表。由于 L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> 的 `clist` 总是会自动忽略空元素，  
\\_\_ctex\_assign\_heading\_name:nn

所以设置 `name={,章}` 后,第一个元素将会是“章”,必须用空的分组保护空元素: `name={ {},章}`,这在使用中有些许不便。我们可以改用 `seq` 或者手写函数解析参数来加以改进。为实现的简单起见,这里用了 `xparse` 的 `\SplitArgument`,它带有参数的长度检查。

```

2732 \NewDocumentCommand \ctex_assign_heading_name:nn
2733   { m > { \SplitArgument { 1 } { , } } +m }
2734   { \_ctex_assign_heading_name:nnn {#1} #2 }
2735 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2736   {
2737     \tl_set:cn { CTEX@pre#1 } {#2}
2738     \tl_if_novalue:nTF {#3}
2739       { \tl_clear:c { CTEX@post#1 } }
2740       { \tl_set:cn { CTEX@post#1 } {#3} }
2741   }

```

`part/pagestyle` 只在 `ctexbook` 和 `ctexrep` 下有定义。  
`chapter/pagestyle`  
`chapter/lofskip`  
`chapter/lotskip`

```

2742 \group_begin:
2743 <*book|report>
2744 \tl_set:Nn \l__ctex_tmp_tl
2745   {
2746     part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2747     chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2748     chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2749     chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2750     chapter / lofskip .initial:n = \c_zero_skip ,
2751     chapter / lotskip .initial:n = \c_zero_skip ,
2752     chapter / lofskip .value_required:n = true ,
2753     chapter / lotskip .value_required:n = true ,
2754   }
2755 </book|report>
2756 <*article|beamer>
2757 \tl_clear:N \l__ctex_tmp_tl
2758 </article|beamer>

```

定义标题键值选项。

```

2759 \seq_map_inline:Nn \c__ctex_headings_seq
2760   {
2761     \_ctex_initial_heading:n {#1}
2762     \_ctex_def_heading_keys:n {#1}
2763   }
2764 \exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }

```

`\CTEX@heading@format@initial` 标题格式的一些初始设置,包括恢复默认字体,并禁用自动调整首行缩进,禁止在标题中分页。同时用 `\noindent` 抑制首行缩进并进入水平模式。统一在各级标题的 `format` 选项之前使用。

```

2765 \cs_new_protected:Npn \CTEX@heading@format@initial
2766   {
2767     \normalfont
2768     \tl_clear:N \l__ctex_autoindent_tl
2769     \int_set:Nn \tex_interlinepenalty:D { 10 000 }
2770     \tex_noindent:D
2771   }

```

`\ctex_indent_box:n` 设置 `\parindent`,并插入用于产生缩进的盒子,如果缩进为0,就不插入。

```

2772 \cs_new_protected:Npn \ctex_indent_box:n #1
2773   {
2774     \dim_set:Nn \tex_parindent:D {#1}
2775     \_ctex_insert_indent:
2776   }
2777 \cs_new_protected:Npn \_ctex_insert_indent:
2778   {

```

```

2779 \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
2780 { \tex_indent:D }
2781 }
2782 \cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n

```

### 15.8.2 标准标题命令的修改

```
2783 <*article|book|report>
```

`\CTEX@fixtopskip` 修正 `book` 和 `report` 类的 `\part` 和 `\chapter` 标题之前的多余空行。

```

2784 <*book|report>
2785 \cs_new_protected:Npn \CTEX@fixtopskip
2786 {
2787 \CTEX@fixheadingskip
2788 \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2789 { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2790 }
2791 </book|report>

```

`\CTEX@fixheadingskip` 抑制行间粘连, 修正标题前后的多余间距。事实上, 减掉 `\parskip`, 有一定的风险。如果接下来的内容不会进入水平模式(例如在 `format` 选项中使用 `\hrule` 或者 `\hbox`),  $\TeX$  就不会加上 `\parskip`。这时候就需要用户把 `\parskip` 加到 `beforekip` 或者 `afterkip` 作为修正。

```

2792 \cs_new_protected:Npn \CTEX@fixheadingskip
2793 {
2794 \par
2795 \dim_set:Nn \tex_prevdepth:D { -1000pt }
2796 \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2797 }
2798 \skip_new:N \l__ctex_heading_skip
2799 \cs_new_protected:Npn \CTEX@setheadingskip
2800 { \skip_set:Nn \l__ctex_heading_skip }
2801 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\CTEX@setthispagestyle` 将 `\CTEX@(part|chapter)@pagestyle` 展开后再传给 `\thispagestyle`。

```

2802 <*book|report>
2803 \cs_new_protected:Npn \CTEX@setthispagestyle #1
2804 { \exp_args:Ne \thispagestyle { \use:c { CTEX@#1@pagestyle } } }
2805 </book|report>

```

`\partmark` 提供 `\partmark`。

```

2806 \ProvideDocumentCommand \partmark { m }
2807 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2808 \cs_new_eq:NN \CTEXifname \use_ii:nn
2809 \cs_new_protected:Npn \CTEX@ifnametrue
2810 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2811 \cs_new_protected:Npn \CTEX@ifnamefalse
2812 { \cs_set_eq:NN \CTEXifname \use_ii:nn }

```

`\CTEX@addloflotskip` 往插图和表格目录中加入额外间距。如果间距为零, 则不加入。

```

2813 <*book|report>
2814 \cs_new_protected:Npn \CTEX@addloflotskip #1
2815 {
2816 \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }

```

```

2817 \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2818 {
2819   \addtocontents { lof }
2820   { \protect \advvspace { \skip_use:N \l__ctex_heading_skip } }
2821 }
2822 \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2823 \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2824 {
2825   \addtocontents { lot }
2826   { \protect \advvspace { \skip_use:N \l__ctex_heading_skip } }
2827 }
2828 }
2829 </book|report>

```

```

\CTEX@addtocline 2830 \cs_new_protected:Npn \CTEX@addtocline #1#2
2831 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

### 15.8.2.1 part 的标题

```

\part 2832 <*article>
2833 \renewcommand\part{%
2834   \if@noskipsec \leavevmode \fi
2835   \par
2836   \CTEX@part@break
2837   % \advvspace{4ex}%
2838   \CTEX@setheadingskip \CTEX@part@beforeskip
2839   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2840   \advvspace \CTEX@headingskip
2841   \ifodd \CTEX@part@afterindent
2842     \@afterindenttrue
2843   \else
2844     \@afterindentfalse
2845   \fi
2846   \secdef\@part\@spart}
2847 </article>
2848 <*book|report>
2849 \renewcommand\part{%
2850 % \if@openright
2851 % \clearpage
2852 % \else
2853 % \clearpage
2854 % \fi
2855 \CTEX@part@break
2856 % \thispagestyle{plain}%
2857 \CTEX@setthispagestyle{part}%
2858 \if@twocolumn
2859   \onecolumn
2860   \@tempwatrue
2861 \else
2862   \@tempwafalse
2863 \fi
2864 % \null\vfil
2865 \CTEX@setheadingskip \CTEX@part@beforeskip
2866 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2867 \vspace*{\CTEX@headingskip}%
2868 \secdef\@part\@spart}
2869 </book|report>

\@part 2870 <*article>
2871 \def\@part [#1]#2{%
2872   \ifnum \c@secnumdepth >\m@ne
2873     \ifodd \CTEX@part@numbering
2874       \CTEX@ifnametrue
2875       \refstepcounter{part}%
2876 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%

```

```

2877 \else
2878 \CTEX@ifnamefalse
2879 \CTEX@makeanchor{part*}%
2880 % \addcontentsline{toc}{part}{#1}%
2881 \fi
2882 \else
2883 \CTEX@ifnamefalse
2884 \CTEX@makeanchor{part*}%
2885 % \addcontentsline{toc}{part}{#1}%
2886 \fi
2887 \CTEX@getttitle{#1}%
2888 \CTEX@addtocline{part}{#1}%
2889 \partmark{#1}%
2890 \beginingroup
2891 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2892 \CTEX@heading@format@initial
2893 \CTEX@part@format{%
2894 % \ifnum \c@secnumdepth >\m@ne
2895 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2896 % \fi
2897 \CTEX@headinghang{part}%
2898 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2899 % \huge\bfseries #2%
2900 \CTEX@part@titleformat{#2}%
2901 % \markboth{}{}\par
2902 \CTEX@part@aftertitle}\par
2903 \endgroup
2904 \nobreak
2905 % \vskip 3ex
2906 \CTEX@setheadingskip \CTEX@part@afterskip
2907 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2908 \vskip \CTEX@headingskip
2909 \@afterheading}
2910 </article>
2911 <*book|report>
2912 \def\@part[#1]#2{%
2913 \ifnum \c@secnumdepth >-2\relax
2914 \ifodd \CTEX@part@numbering
2915 \CTEX@ifnametrue
2916 \refstepcounter{part}%
2917 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2918 \else
2919 \CTEX@ifnamefalse
2920 \CTEX@makeanchor{part*}%
2921 % \addcontentsline{toc}{part}{#1}%
2922 \fi
2923 \else
2924 \CTEX@ifnamefalse
2925 \CTEX@makeanchor{part*}%
2926 % \addcontentsline{toc}{part}{#1}%
2927 \fi
2928 \CTEX@getttitle{#1}%
2929 \CTEX@addtocline{part}{#1}%
2930 % \markboth{}{}\par
2931 \partmark{#1}%
2932 \beginingroup
2933 % \centering \interlinepenalty \@M \normalfont
2934 \CTEX@heading@format@initial
2935 \CTEX@part@format{%
2936 % \ifnum \c@secnumdepth >-2\relax
2937 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2938 % \fi
2939 \CTEX@headinghang{part}%
2940 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2941 % \huge\bfseries #2\par
2942 \CTEX@part@titleformat{#2}%
2943 \CTEX@part@aftertitle}\par

```

```

2944 \endgroup
2945 \@endpart}
2946 </book|report>

\@spart 2947 <*article>
2948 \def\@spart#1{%
2949 \CTEX@ifnamefalse
2950 \CTEX@makeanchor@spart{part*}%
2951 \CTEX@getttitle{#1}%
2952 \beginngroup
2953 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2954 \CTEX@heading@format@initial
2955 \CTEX@part@format{%
2956 \CTEX@headinghang{part}{}%
2957 % \huge \bfseries #1\par
2958 \CTEX@part@titleformat{#1}%
2959 \CTEX@part@aftertitle}\par
2960 \endgroup
2961 \nobreak
2962 % \vskip 3ex
2963 \CTEX@setheadingskip \CTEX@part@afterskip
2964 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2965 \vskip \CTEX@headingskip
2966 \@afterheading}
2967 </article>
2968 <*book|report>
2969 \def\@spart#1{%
2970 \CTEX@ifnamefalse
2971 \CTEX@makeanchor@spart{part*}%
2972 \CTEX@getttitle{#1}%
2973 \beginngroup
2974 % \centering \interlinepenalty \@M \normalfont
2975 \CTEX@heading@format@initial
2976 \CTEX@part@format{%
2977 \CTEX@headinghang{part}{}%
2978 % \Huge \bfseries #1\par%
2979 \CTEX@part@titleformat{#1}%
2980 \CTEX@part@aftertitle}\par
2981 \endgroup
2982 \@endpart}
2983 </book|report>

\@endpart 2984 <*book|report>
2985 \def\@endpart{%
2986 % \vfil
2987 \CTEX@setheadingskip \CTEX@part@afterskip
2988 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2989 \vskip \CTEX@headingskip
2990 \newpage
2991 \if@twoside
2992 \if@openright
2993 \null
2994 \thispagestyle{empty}%
2995 \newpage
2996 \fi
2997 \fi
2998 \if@tempwa
2999 \twocolumn
3000 \fi}
3001 </book|report>

15.8.2.2 chapter 的标题
3002 <*book|report>

\chapter 3003 \renewcommand\chapter{%
3004 % \if@openright\cleardoublepage\else\clearpage\fi

```

```

3005 % \thispagestyle{plain}%
3006 \CTEX@chapter@break
3007 \CTEX@setthispagestyle{chapter}%
3008 \global\@topnum\z@
3009 % \@afterindentfalse
3010 \ifodd \CTEX@chapter@afterindent
3011     \@afterindenttrue
3012 \else
3013     \@afterindentfalse
3014 \fi
3015 \secdef\@chapter\@schapter}

\@chapter 3016 \def\@chapter[#1]#2{%
3017     \ifnum \c@secnumdepth >\m@ne
3018 <*book>
3019     \if@mainmatter
3020 </book>
3021         \ifodd \CTEX@chapter@numbering
3022             \CTEX@ifnametrue
3023             \refstepcounter{chapter}%
3024 %             \typeout{\@chapapp\space\thechapter.}%
3025             \typeout{\CTEXthechapter}%
3026 %             \addcontentsline{toc}{chapter}
3027 %             {\protect\numberline{\thechapter}#1}%
3028         \else
3029             \CTEX@ifnamefalse
3030             \CTEX@makeanchor{\Hy@chapapp*}%
3031 %             \addcontentsline{toc}{chapter}{#1}%
3032         \fi
3033 <*book>
3034         \else
3035             \CTEX@ifnamefalse
3036             \CTEX@makeanchor@chapter{\Hy@chapapp*}%
3037 %             \addcontentsline{toc}{chapter}{#1}%
3038         \fi
3039 </book>
3040         \else
3041             \CTEX@ifnamefalse
3042             \CTEX@makeanchor@chapter{\Hy@chapapp*}%
3043 %             \addcontentsline{toc}{chapter}{#1}%
3044         \fi
3045         \CTEX@getttitle{#1}%
3046         \CTEX@addtoctline{chapter}{#1}%
3047         \chaptermark{#1}%
3048 % \addtocontents{lof}{\protect\advspace{10\p@}}%
3049 % \addtocontents{lot}{\protect\advspace{10\p@}}%
3050         \CTEX@addloflotskip{chapter}%
3051         \if@twocolumn
3052             \@topnewpage[\@makechapterhead{#2}]%
3053         \else
3054             \@makechapterhead{#2}%
3055         \@afterheading
3056         \fi}

\@schapter 3057 \def\@schapter#1{%
3058     \CTEX@ifnamefalse
3059     \CTEX@makeanchor@schapter{\Hy@chapapp*}%
3060     \CTEX@getttitle{#1}%
3061     \if@twocolumn
3062         \@topnewpage[\@makeschapterhead{#1}]%
3063     \else
3064         \@makeschapterhead{#1}%
3065     \@afterheading
3066     \fi}

\@makechapterhead 3067 \def\@makechapterhead#1{%
3068 % \vspace*{50\p@}%

```

```

3069 \CTEX@setheadingskip \CTEX@chapter@beforeskip
3070 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
3071 \vspace*{\CTEX@headingskip}%
3072 \begingroup
3073 % \parindent \z@ \raggedright \normalfont
3074 \CTEX@heading@format@initial
3075 \CTEX@chapter@format{%
3076 % \ifnum \c@secnumdepth >\m@ne
3077 % \if@mainmatter
3078 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
3079 % \fi
3080 % \fi
3081 \CTEX@headinghang{chapter}%
3082 {\CTEXifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
3083 % \Huge \bfseries #1\par\nobreak
3084 \CTEX@chapter@titleformat{#1}%
3085 \CTEX@chapter@aftertitle}\par
3086 \endgroup
3087 \nobreak
3088 % \vskip 40\p@
3089 \CTEX@setheadingskip \CTEX@chapter@afterskip
3090 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
3091 \vskip \CTEX@headingskip}

\@makeschapterhead 3092 \def\@makeschapterhead#1{%
3093 % \vspace*{50\p@}%
3094 \CTEX@setheadingskip \CTEX@chapter@beforeskip
3095 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
3096 \vspace*{\CTEX@headingskip}%
3097 \begingroup
3098 % \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
3099 \CTEX@heading@format@initial
3100 \CTEX@chapter@format{%
3101 \CTEX@headinghang{chapter}{}}%
3102 % \Huge \bfseries #1\par\nobreak
3103 \CTEX@chapter@titleformat{#1}%
3104 \CTEX@chapter@aftertitle}\par
3105 \endgroup
3106 \nobreak
3107 % \vskip 40\p@
3108 \CTEX@setheadingskip \CTEX@chapter@afterskip
3109 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
3110 \vskip \CTEX@headingskip}

3111 </book|report>

```

### 15.8.2.3 section 类的标题

\startsection L<sup>A</sup>T<sub>E</sub>X 的标准参数是:

```

{\name}{level}{indent}{beforeskip}{afterskip}{style}* [altheading]{heading}

3112 \def\startsection#1#2#3#4#5#6{%
3113 \if@noskipsec \leavevmode \fi
3114 \par
3115 % \@tempskipa #4\relax
3116 % \@afterindenttrue
3117 % \ifdim \@tempskipa <\z@
3118 % \@tempskipa -\@tempskipa \@afterindentfalse
3119 % \fi
3120 \CTEX@update@sectionformat@n{#1}%
3121 \ifodd \CTEX@afterindent
3122 \@afterindenttrue
3123 \else
3124 \@afterindentfalse
3125 \fi
3126 \if@nobreak

```

```

3127 \everypar{%
3128 \else
3129 % \addpenalty\@secpenalty\addvspace\@tempskipa
3130 \csname CTEX@#1@break\endcsname
3131 \CTEX@setheadingskip{#4}%
3132 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
3133 \addvspace \CTEX@headingskip
3134 \fi
3135 \@ifstar
3136 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
3137 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

\@secntformat 3138 \def\@secntformat#1{%
3139 % \csname the#1\endcsname\quad}%
3140 \csname CTEX@#1name\endcsname
3141 \csname CTEX@#1@aftername\endcsname}

\@sect 3142 \def\@sect#1#2#3#4#5#6[#7]#8{%
3143 \ifnum #2>\c@secnumdepth
3144 \CTEX@ifnamefalse
3145 \CTEX@makeanchor@ssect{#1*}%
3146 \let\@svsec\@empty
3147 \else
3148 \ifodd \csname CTEX@#1@numbering\endcsname
3149 \CTEX@ifnametrue
3150 \refstepcounter{#1}%
3151 \protected@edef\@svsec{\@secntformat{#1}\relax}%
3152 \else
3153 \CTEX@ifnamefalse
3154 \CTEX@makeanchor{#1*}%
3155 \let\@svsec\@empty
3156 \fi
3157 \fi
3158 \CTEX@getttitle{#7}%
3159 % \@tempskipa #5\relax
3160 % \ifdim \@tempskipa>\z@
3161 \unless \ifodd \CTEX@runin
3162 \begingroup
3163 \CTEX@heading@format@initial
3164 #6{%
3165 % \@hangfrom{\hskip #3\relax\@svsec}%
3166 % \interlinepenalty \@M #8\@@par
3167 \CTEX@sectionhang{#3}{\@svsec}%
3168 \csname CTEX@#1@titleformat\endcsname{#8}%
3169 \csname CTEX@#1@aftertitle\endcsname}\par
3170 \endgroup
3171 \csname #1mark\endcsname{#7}%
3172 % \addcontentsline{toc}{#1}{%
3173 % \ifnum #2>\c@secnumdepth \else
3174 % \protect\numberline{\csname the#1\endcsname}%
3175 % \fi
3176 % #7}%
3177 \CTEX@addtocline{#1}{#7}%
3178 \else
3179 \def\@svsechd{%
3180 #6{%
3181 % \hskip #3\relax \@svsec #8
3182 {\CTEX@indentbox{#3}}\@svsec
3183 \csname CTEX@#1@titleformat\endcsname{#8}%
3184 \csname CTEX@#1@aftertitle\endcsname}%
3185 \csname #1mark\endcsname{#7}%
3186 % \addcontentsline{toc}{#1}{%
3187 % \ifnum #2>\c@secnumdepth \else
3188 % \protect\numberline{\csname the#1\endcsname}%
3189 % \fi
3190 % #7}%
3191 \CTEX@addtocline{#1}{#7}}%

```

```

3192 \fi
3193 \@xsect{#5}}

\@ssect 3194 \def\@ssect#1#2#3#4#5{%
3195   \CTEX@ifnamefalse
3196   \CTEX@getttitle{#5}%
3197   % \@tempskipa #3\relax
3198   % \ifdim \@tempskipa>\z@
3199   \unless \ifodd \CTEX@runin
3200     \begingroup
3201       \CTEX@heading@format@initial
3202       #4{%
3203         \@hangfrom{\hskip #1}%
3204         \interlinepenalty \@M #5\__ctexpar
3205         \CTEX@sectionhang{#1}{}%
3206         \CTEX@titleformat@n{#5}%
3207         \CTEX@aftertitle}\par
3208       \endgroup
3209   \else
3210   % \def\@svsechd{#4{\hskip #1\relax #5}}%
3211   \def\@svsechd{#4{\CTEX@indentbox{#1}}%
3212     \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
3213   \fi
3214   \@xsect{#3}}

\@xsect 3215 \def\@xsect#1{%
3216   % \@tempskipa #1\relax
3217   % \ifdim \@tempskipa>\z@
3218   \unless \ifodd \CTEX@runin
3219     \par \nobreak
3220   % \vskip \@tempskipa
3221   \CTEX@setheadingskip{#1}%
3222   \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
3223   \vskip \CTEX@headingskip
3224   \@afterheading
3225   \else
3226     \@nobreakfalse
3227     \global\@noskipsectrue
3228     \everypar{%
3229       \if@noskipsec
3230         \global\@noskipsecfalse
3231         {\setbox\z@\lastbox}%
3232         \clubpenalty\@M
3233         \begingroup \@svsechd \endgroup
3234         \unskip
3235       % \@tempskipa #1\relax
3236       % \hskip -\@tempskipa
3237       \CTEX@heading@glue{#1}%
3238     \else
3239       \clubpenalty \@clubpenalty
3240       \everypar{}}%
3241     \fi}%
3242   \fi
3243   \ignorespaces}

\CTEX@headinghang 分别用于实现 \part/\chapter 和 \section 类标题的 indent 和 hang 选项。
\CTEX@sectionhang
3244 \cs_new_protected:Npn \CTEX@headinghang #1
3245 {
3246   \ctex_heading_hang:cnn
3247   { \CTEX@#1@hang }
3248   { \use:c { \CTEX@#1@indent } }
3249 }
3250 \cs_new_protected:Npn \CTEX@sectionhang
3251 { \ctex_heading_hang:Nnn \CTEX@hang }

```

`\ctex_heading_hang:Nnn` `hang` 选项控制是否采用悬挂缩进,同时设置 `\parindent`。  
`\ctex_hang_from:n`

```

3252 \cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
3253 {
3254   \dim_set:Nn \tex_parindent:D {#2}
3255   \bool_if:NTF #1
3256     { \ctex_hang_from:n }
3257     { \use:n }
3258     { \__ctex_insert_indent: #3 }
3259 }
3260 \cs_new_protected:Npn \ctex_hang_from:n #1
3261 {
3262   \tex_noindent:D
3263   \hbox_set:Nn \l__ctex_tmp_box {#1}
3264   \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
3265   \box_use_drop:N \l__ctex_tmp_box
3266 }
3267 \cs_generate_variant:Nn \ctex_heading_hang:Nnn { c }

```

`\ctex_heading_glue:n` 如果缩进#1 长度为零,就不插入水平间距。  
`\CTEX@heading@glue`

```

3268 \cs_new_protected:Npn \ctex_heading_glue:n #1
3269 {
3270   \group_begin:
3271     \skip_set:Nn \l__ctex_heading_skip {#1}
3272     \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
3273       { \skip_horizontal:N \l__ctex_heading_skip }
3274   \group_end:
3275 }
3276 \cs_new_eq:NN \CTEX@heading@glue \ctex_heading_glue:n

```

`\CTEX@update@sectionformat@n` 在 `\@startsection` 中设置 `\CTEX@titleformat@n` 等为相应函数。

```

3277 \cs_new_protected:Npn \CTEX@update@sectionformat@n #1
3278 {
3279   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
3280   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
3281   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
3282   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
3283   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
3284   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
3285 }
3286 \cs_new_eq:NN \CTEX@titleformat@n \use:n
3287 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
3288 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
3289 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
3290 \cs_new_eq:NN \CTEX@hang \c_true_bool
3291 \cs_new_eq:NN \CTEX@runin \c_false_bool

```

```

\CTEX@part@tocline 3292 \cs_new:Npn \CTEX@part@tocline #1#2
\CTEX@chapter@tocline 3293 {
3294   \CTEXifname
3295     { \CTEXthepart \hspace { 1em } }
3296     { }
3297   #2
3298 }
3299 <*book|report>
3300 \cs_new:Npn \CTEX@chapter@tocline #1#2
3301 {
3302   \CTEXifname
3303     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
3304     { }
3305   #2
3306 }
3307 </book|report>

```

`\CTEXnumberline` 3308 `\cs_new:Npn \CTEXnumberline #1`

```

3309 {
3310   \CTEXifname
3311   { \protect \numberline { \use:c { CTEXthe #1 } } }
3312   { }
3313 }

3314 \int_zero:N \l__ctex_tmp_int
3315 \seq_map_inline:Nn \c__ctex_section_headings_seq
3316 {
3317   \int_incr:N \l__ctex_tmp_int
3318   \cs_gset_protected:cpx {#1}
3319   {
3320     \exp_not:N \@startsection {#1}
3321     { \int_use:N \l__ctex_tmp_int }
3322     { \exp_not:c { CTEX@#1@indent } }
3323     { \exp_not:c { CTEX@#1@beforeskip } }
3324     { \exp_not:c { CTEX@#1@afterskip } }
3325     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
3326   }
3327   \cs_new:cpx { CTEX@#1@tocline } ##1##2
3328   { \CTEXnumberline { ##1 } ##2 }
3329 }

```

#### 15.8.2.4 附录标题

```

appendix/name 3330 \ctex_define:n
appendix/number 3331 {
appendix/numbering 3332   appendix .meta:nn = { ctex / appendix } {#1} ,
3333   appendix / name .code:n =
3334   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
3335   appendix / number .tl_set:N = \CTEX@appendix@number ,
3336   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3337   appendix / numbering .initial:n = true
3338 }
3339 \tl_new:N \CTEX@preappendix
3340 \tl_new:N \CTEX@postappendix

\appendix 3341 \cs_new_eq:NN \CTEX@save@appendix \appendix
3342 \cs_gset_protected:Npn \appendix
3343 {
3344   \CTEX@save@appendix
3345   <*article>
3346   \gdef \CTEX@presection { \CTEX@preappendix }
3347   \gdef \CTEX@thesection { \CTEX@appendix@number }
3348   \gdef \CTEX@postsection { \CTEX@postappendix }
3349   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3350   </article>
3351   <*book|report>
3352   \gdef \CTEX@prechapter { \CTEX@preappendix }
3353   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3354   \gdef \CTEX@postchapter { \CTEX@postappendix }
3355   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3356   </book|report>
3357 }

```

#### 15.8.2.5 设置 hyperref 宏包的标题锚点

\CTEX@makeanchor 设置超链接跳转锚点, 在 hyperref 载入后才有意义。

```

3358 \cs_new_protected:Npn \CTEX@makeanchor #1
3359 { }

```

\c\_\_ctex\_headings\_cs\_seq 保存内部标题命令的 CT<sub>E</sub>X 定义, 用于随后比较。

```

3360 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq

```

```

<article> 3361 { part , spart , sect , ssect }
<book|report> 3362 { part , spart , chapter , schapter , sect , ssect }
3363 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3364 {
3365   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3366   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3367 }

```

`\CTEX@hyperheadinghook` `hyperref` 会重定义内部标题命令，目的在于为没有编号的标题设置锚点（这一功能受他的 `implicit` 选项的控制）。我们在上面对标题命令的修改已经包含这一功能，如果这些标题命令在 `hyperref` 载入之前没有被修改过，则恢复 `CTEX` 的定义。

```

3368 \cs_new_protected:Npn \CTEX@hyperheadinghook
3369 {
3370   \group_begin:
3371     \legacy_if:nTF { Hy@implicit }
3372     {
3373       \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3374       \seq_map_inline:Nn \c__ctex_headings_cs_seq
3375       {
3376         \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3377         {
3378           \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3379           \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3380         }
3381       }
3382     }
3383     {
3384       \seq_map_inline:Nn \c__ctex_headings_cs_seq
3385       { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3386     }
3387   \group_end:
3388 }

3389 \ctex_at_end_package:nn { hyperref }
3390 {
3391   \cs_gset_protected:Npn \CTEX@makeanchor #1
3392   {
3393     \Hy@MakeCurrentHrefAuto {#1}
3394     \Hy@raisedlink
3395     {
3396       \hyper@anchorstart { \@currentHref }
3397       \hyper@anchorend
3398     }
3399   }
3400   \CTEX@hyperheadinghook
3401 }

```

### 15.8.2.6 兼容 `nameref` 宏包

`\CTEX@getttitle` 在 `nameref` 载入后才有意义，与上述 `hyperref` 的处理类似。

```

3402 \cs_new_protected:Npn \CTEX@getttitle #1
3403 { }
3404 \ctex_at_end_package:nn { nameref }
3405 {
3406   \cs_gset_protected:Npn \CTEX@getttitle { \NR@getttitle }
3407   \seq_map_inline:Nn \c__ctex_headings_cs_seq
3408   {
3409     \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3410     { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3411   }
3412 }

```

## 15.8.2.7 兼容 titlesec 宏包

我们修改了 `\@startsection` 的定义, 它的第四个 (`\before`) 和第五个 (`\after`) 参数的符号不再有特殊意义, 改由相应的选项 `afterindent` 和 `runin` 来控制。

引入 `titlesec` 宏包, 并且未设置它的 `loadonly` 选项时, `titlesec` 会展开 `section` 类标题获取它们的参数, 进行初始设置。我们需要进行一些调整。

`\ctex_titlesec_hook`: `\titleformat` 的设置保存在名为 `\ttl@<section>` 的宏中备用, 它的内容是

```
\ttl@<shape>{<format>}{<label>}{<sep>}{<before>}{<after>}
```

我们这里的 `<shape>` 为 `hang` 或者 `runin`。 `\titlespacing` 的设置保存在 `\ttls@<section>` 之中, 它的内容是

```
{<left>}{<right>}{<before>}{<after>}{<afterindent>}
```

其中 `<afterindent>` 为 1 或 0, 分别对应是否保留段首缩进。我们需要根据 C<sub>T</sub>E<sub>X</sub> 的 `runin` 和 `afterindent` 选项调整 `\ttl@<shape>` 和 `<afterindent>`。注意, 由 `\ttl@extract` 得的 `<before>` 和 `<after>` 的值总是非负的, 而 C<sub>T</sub>E<sub>X</sub> 的 `before` 和 `after` 是可以取负值的, 但我们不打算调整它们了。如果使用了 `titlesec` 的 `indentafter` 等选项, 也不需要调整 `\ttls@<section>`。

```
3413 \cs_new_protected:Npn \ctex_titlesec_hook:
3414 {
3415   \@ifpackagewith { titlesec } { explicit }
3416   {
3417     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3418     \__ctex_titlesec_format_explicit:Nn
3419   }
3420   {}
3421   \clist_map_inline:nn
3422   { indentafter , noindentafter , indentfirst , nonindentfirst }
3423   {
3424     \@ifpackagewith { titlesec } { ##1 }
3425     {
3426       \clist_map_break:n
3427       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3428     }
3429     {}
3430   }
3431   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3432 }
3433 \cs_new_protected:Npn \__ctex_titlesec_hook:n #1
3434 {
3435   \__ctex_titlesec_format:n {#1}
3436   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3437 }
3438 \cs_new_protected:Npn \__ctex_titlesec_format:n #1
3439 {
3440   \cs_if_free:cF { ttl@#1 }
3441   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttl@#1 } {#1} }
3442 }
3443 \cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
3444 {
3445   \tl_set:Ne #1
3446   {
3447     \bool_if:cTF { CTEX@#2@runin }
3448     { \exp_not:N \ttl@runin }
3449     { \exp_not:N \ttl@hang }
3450     \tl_tail:N #1
3451   }
3452 }
```

```

3453 \cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3454 {
3455   \cs_set_nopar:Npx #1 ##1
3456   {
3457     \bool_if:cTF { CTEX@#2@runin }
3458     { \exp_not:N \ttlh@runin }
3459     { \exp_not:N \ttlh@hang }
3460     \exp_args:No \tl_tail:n { #1 { } }
3461   }
3462 }
3463 \cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
3464 { \tl_set:Ne #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3465 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3466 {
3467   \exp_not:n { {#1} {#2} {#3} {#4} }
3468   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3469 }

3470 \@ifpackageloaded { titlesec }
3471 { }
3472 {
3473   \ctex_at_end_package:nn { titlesec }
3474   {
3475     \@ifpackagewith { titlesec } { loadonly }
3476     { }
3477     { \ctex_titlesec_hook: }
3478   }
3479 }

```

让编译时终端显示 `\CTEXthechapter`, 目录使用 `\CTEXtheXXX` 编号。

```

3480 \ctex_at_end_package:nn { titlesec }
3481 {
3482   <*book|report>
3483   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3484   </book|report>
3485   \cs_if_free:NF \ttl@tocpart
3486   {
3487     \cs_set_protected:Npn \ttl@tocpart
3488     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3489   }
3490   \seq_map_inline:Nn \c__ctex_headings_seq
3491   {
3492     \cs_if_exist:cF { \ttl@toc #1 }
3493     {
3494       \cs_new_protected:cpx { \ttl@toc #1 }
3495       {
3496         \tl_set:Nn \exp_not:N \ttl@a
3497         {
3498           \exp_not:N \protect
3499           \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
3500         }
3501       }
3502     }
3503   }
3504 }

```

在 `titlesec` 包定义的标题中更新 `\CTEXifname`。

```

3505 \ctex_at_end_package:nn { titlesec }
3506 {
3507   \ctex_patch_cmd:Nnn \ttl@labelling
3508   { \let \ifttl@toclabel \ifttl@label }
3509   {
3510     \let \ifttl@toclabel \ifttl@label
3511     \CTEX@updatettlifname
3512   }

```

```

3513 \cs_new_protected:Npn \CTEX@updatettlifname
3514 { \legacy_if:nTF { ttl@label } { \CTEX@ifnametrue } { \CTEX@ifnamefalse } }
3515 }

```

### 15.8.2.8 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题<sup>19</sup>。

\ctex\_titleps\_hook: 我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```

3516 \group_begin:
3517 \char_set_catcode_other:N \#
3518 \cs_new_protected:Npn \ctex_titleps_hook:
3519 {
3520   \ctex_patch_cmd:Nnn \ttl@settopmark
3521   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3522   {
3523     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3524     \CTEX@titlepslabel@set {#1}
3525   }
3526   \ctex_patch_cmd:Nnn \ttl@setsubmark
3527   { \protect \@namedef { the#1 } { } }
3528   {
3529     \protect \@namedef { the#1 } { }
3530     \CTEX@titlepslabel@clear {#1}
3531   }
3532   \ctex_patch_cmd:Nnn \ttl@setsubmark
3533   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3534   {
3535     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3536     \CTEX@titlepslabel@set {#2}
3537   }
3538 }
3539 \group_end:

```

\CTEX@titlepslabel@set  
\CTEX@titlepslabel@clear 这两个函数要在随后被 \xdef 展开来获得 \CTEXtheXXX 的内容, 不应该用 \protected 来定义。

```

3540 \cs_new:Npn \CTEX@titlepslabel@set #1
3541 {
3542   \cs_if_free:cF { CTEXthe#1 }
3543   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3544 }
3545 \cs_new:Npn \CTEX@titlepslabel@clear #1
3546 {
3547   \cs_if_free:cF { CTEXthe#1 }
3548   { \protect \@namedef { CTEXthe#1 } { } }
3549 }

```

titleps 宏包的功能可以由 titlesec 的选项 pagestyles 引入。

```

3550 \ctex_at_end_package:nn { titlesec }
3551 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3552 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }

```

除此之外, 也可以使用 titleps 提供的命令 \newtitlemark 来完成:

```

\newtitlemark { \CTEXthechapter }
\newtitlemark { \CTEXthesection }

```

<sup>19</sup><https://github.com/CTeX-org/ctex-kit/issues/217>

但 `\newtitlemark` 不包含章节间的层次信息, 功能上不及修改内部命令完整。

`\ttl@setifthe` 使 `\iftheXXX` 等命令在页眉设置中可用。

```

3553 \ctex_at_end_package:nn { titles }
3554 {
3555   \cs_set_protected:Npn \ttl@setifthe #1
3556   {
3557     \exp_args:Nco \cs_set:Npn { ifthe #1 }
3558     {
3559       \CTEXifname
3560       { \protect \@firstoftwo }
3561       { \protect \@secondoftwo }
3562     }
3563   }
3564   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3565 }

```

### 15.8.3 目录标签的宽度

```

\CTEX@toc@width@n 3566 \cs_new_protected:Npn \CTEX@toc@width@n #1
3567 {
3568   \hbox_set:Nn \l__ctex_tmp_box {#1}
3569   \dim_set:Nn \@tempdima
3570   {
3571     \dim_max:nn { \@tempdima }
3572     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3573   }
3574 }

```

`\numberline` 为 `\numberline` 命令打补丁, 并兼容 `tocloft` 和 `titletoc` 宏包。

`\_ctex_patch_toc_width:n`

这里需要替换# 本身, 因此需要先切换为 `other` 类。表示参数的# 用 `\c_parameter_token` 代替。

```

3575 \group_begin:
3576 \char_set_catcode_other:N \#
3577 \use:n
3578 {
3579   \group_end:
3580   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3581   { \CTEX@toc@width@n {#1} }
3582   { }
3583   { \ctex_patch_failure:N \numberline }
3584   \cs_new_protected:Npn \_ctex_patch_toc_width:n \c_parameter_token 1
3585   {
3586     \ifpackageloaded { \c_parameter_token 1 }
3587     {
3588       {
3589         \ctex_at_end_package:nn { \c_parameter_token 1 }
3590         {
3591           \ctex_preto_cmd:NnnTF \numberline
3592           { \char_set_catcode_letter:n { 64 } }
3593           { \CTEX@toc@width@n {#1} }
3594           { }
3595           { \ctex_patch_failure:N \numberline }
3596         }
3597       }
3598     }
3599   }
3600   \_ctex_patch_toc_width:n { tocloft }
3601   \_ctex_patch_toc_width:n { titletoc }

```

## 15.8.4 页眉信息的修改

```

\ps@headings 3602 <*article>
3603 \legacy_if:nTF { @twoside }
3604 {
3605   \ctex_patch_cmd:Nnn \ps@headings
3606     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3607     { \CTEXifname { \CTEXthesection \quad } { } }
3608   \ctex_patch_cmd:Nnn \ps@headings
3609     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3610     { \CTEXifname { \CTEXthesubsection \quad } { } }
3611 }

```

不知为何,标准文档类此处对 `secnumdepth` 的判断为 0,与 `\section` 的层次 1 不符。

```

3612 {
3613   \ctex_patch_cmd:Nnn \ps@headings
3614     { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3615     { \CTEXifname { \CTEXthesection \quad } { } }
3616 }
3617 </article>
3618 <*book|report>
3619 \ctex_patch_cmd:Nnn \ps@headings
3620 {
<book> 3621   \ifnum \c@secnumdepth > \m@ne \if@mainmatter
<report> 3622   \ifnum \c@secnumdepth > \m@ne
3623     \@chapapp \ \thechapter . ~ \ %
<report> 3624   \fi
<book> 3625   \fi \fi
3626 }
3627 { \CTEXifname { \CTEXthechapter \quad } { } }
3628 \legacy_if:nTF { @twoside }
3629 {
3630   \ctex_patch_cmd:Nnn \ps@headings
3631     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3632     { \CTEXifname { \CTEXthesection \quad } { } }
3633 }
3634 </book|report>

```

`\f@nch@initialise` 这里对 `fancyhdr` 宏包打补丁。原来 `fancyhdr` 宏包中使用 `\thesection` 等宏表示页眉中的章节编号,这里改用 `ctex` 包所用的 `\CTEXthesection` 系列宏。

```

3635 \ctex_at_end_package:nn { fancyhdr }
3636 {
3637   \ctex_patch_cmd:Nnn \f@nch@initialise
3638     { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3639     { \CTEXifname { \CTEXthesection \quad } { } }
3640   \ctex_patch_cmd:Nnn \f@nch@initialise
3641     { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3642     { \CTEXifname { \CTEXthesubsection \quad } { } }
3643   \ctex_patch_cmd:Nnn \f@nch@initialise
3644     { \ifnum \c@secnumdepth > \m@ne \@chapapp \ \thechapter . ~ \ \fi }
3645     { \CTEXifname { \CTEXthechapter \quad } { } }
3646   \ctex_patch_cmd:Nnn \f@nch@initialise
3647     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3648     { \CTEXifname { \CTEXthesection \quad } { } }
3649   \f@nch@initialise

```

`fancyhdr` 的 `headings` 选项会重定义 `\ps@headings`,这里也要打补丁。

```

3650   \@ifpackagewith { fancyhdr } { headings }
3651   {
3652 <*article>
3653     \legacy_if:nTF { @twoside }
3654     {
3655       \ctex_patch_cmd:Nnn \ps@headings
3656         { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }

```

```

3657         { \CTEXifname { \CTEXthesection \quad } { } }
3658     \ctex_patch_cmd:Nnn \ps@headings
3659     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3660     { \CTEXifname { \CTEXthesubsection \quad } { } }
3661     }
3662     {
3663     \ctex_patch_cmd:Nnn \ps@headings
3664     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3665     { \CTEXifname { \CTEXthesection \quad } { } }
3666     }
3667 </article>
3668 <*book|report>
3669     \ctex_patch_cmd:Nnn \ps@headings
3670     {
3671     \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3672     \ifnum \c@secnumdepth > \m@ne
3673     \@chapapp \ thechapter . ~ \ %
3674     \fi
3675     \fi \fi
3676     }
3677     { \CTEXifname { \CTEXthechapter \quad } { } }
3678     \legacy_if:nT { @twoside }
3679     {
3680     \ctex_patch_cmd:Nnn \ps@headings
3681     { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3682     { \CTEXifname { \CTEXthesection \quad } { } }
3683     }
3684 </book|report>
3685     }
3686     { }
3687     }
3688 </article|book|report>

```

### 15.8.5 beamer 标题页模板的修改

```

3689 <*beamer>
3690 \ExplSyntaxOff

```

对应 \partpage。

```

3691 \defbeamer*template*{part page}{CTEX}[1][ ]{%
3692 \begingroup
3693 % \centering
3694 % {\usebeamerfont{part name}%
3695 % \usebeamerfont{part name}\partname~\insertromanpartnumber}
3696 % \vskip1em\par
3697 \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3698 \CTEX@heading@format@initial
3699 \CTEX@part@format{%
3700 \CTEX@indentbox{\CTEX@part@indent}%
3701 \ifodd \CTEX@part@numbering
3702 \CTEX@partname \CTEX@part@aftername
3703 \fi
3704 \begin{beamer*colorbox}[sep=16pt,center,#1]{part title}
3705 % \usebeamerfont{part title}\insertpart\par
3706 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3707 \end{beamer*colorbox}}%
3708 \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3709 \endgroup}

```

对应 \sectionpage。

```

3710 \defbeamer*template*{section page}{CTEX}[1][ ]{%
3711 \begingroup
3712 % \centering
3713 % {\usebeamerfont{section name}%

```

```

3714 % \usebeamerfont[fg]{section name}\sectionname~\insertsectionnumber}
3715 % \vskip1em\par
3716 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3717 \CTEX@heading@format@initial
3718 \CTEX@section@format{%
3719 \CTEX@indentbox{\CTEX@section@indent}%
3720 \ifodd \CTEX@section@numbering
3721 \CTEX@sectionname \CTEX@section@aftername
3722 \fi
3723 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3724 % \usebeamerfont{section title}\insertsection\par
3725 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3726 \end{beamercolorbox}}%
3727 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3728 \endgroup}

```

对应 \subsectionpage。

```

3729 \defbeamerfont*{subsection page}{CTEX}[1] [] {%
3730 \begin{group}
3731 % \centering
3732 % {\usebeamerfont{subsection name}}%
3733 % \usebeamerfont[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3734 % \vskip1em\par
3735 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3736 \CTEX@heading@format@initial
3737 \CTEX@subsection@format{%
3738 \CTEX@indentbox{\CTEX@subsection@indent}%
3739 \ifodd \CTEX@subsection@numbering
3740 \CTEX@subsectionname \CTEX@subsection@aftername
3741 \fi
3742 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3743 % \usebeamerfont{subsection title}\insertsubsection\par
3744 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3745 \end{beamercolorbox}}%
3746 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3747 \endgroup}

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3748 \defbeamerfontalias{part page}{default}{CTEX}
3749 \defbeamerfontalias{section page}{default}{CTEX}
3750 \defbeamerfontalias{subsection page}{default}{CTEX}
3751 \ExplSyntaxOn
3752 </beamer>

```

### 15.8.6 标题编号和目录的层次设置

**secnumdepth** secnumdepth 在 beamer 下无意义。  
**tocdepth**

```

3753 \ctex_define:n
3754 {
3755 <!*beamer>
3756 secnumdepth .code:n = \ctex_heading_depth:ne { secnumdepth } {#1} ,
3757 secnumdepth .value_required:n = true ,
3758 </!beamer>
3759 tocdepth .code:n = \ctex_heading_depth:ne { tocdepth } {#1} ,
3760 tocdepth .value_required:n = true
3761 }

```

\ctex\_heading\_depth:nn 注意此处 \setcounter 的赋值是全局的。

```

3762 \cs_new_protected:Npn \ctex_heading_depth:nn #1#2
3763 {
3764 \prop_get:NnNTF \c__ctex_heading_level_prop {#2} \l__ctex_tmp_tl
3765 { \setcounter {#1} { \l__ctex_tmp_tl } }

```

```

3766     { \setcounter {#1} { \int_eval:n {#2} } }
3767   }
3768 \cs_generate_variant:Nn \ctex_heading_depth:nn { ne }

```

`\c__ctex_heading_level_prop` 章节层次与名称的对应表。

```

3769 \prop_const_from_keyval:Nn \c__ctex_heading_level_prop
3770   {
3771   <*article|beamer>
3772     part = 0 ,
3773   </article|beamer>
3774   <*book|report>
3775     part = -1 ,
3776     chapter = 0 ,
3777   </book|report>
3778     section = 1 ,
3779     subsection = 2 ,
3780     subsubsection = 3 ,
3781     paragraph = 4 ,
3782     subparagraph = 5
3783   }

```

### 15.8.7 标签引用数字的汉化

`\refstepcounter` 对标题进行引用时, 设置标签为通过 `number` 选项设置的形式。

```

3784 \cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
3785   {
3786     \protected@edef \@currentlabel
3787       {
3788         \cs_if_exist:cTF { CTEX@the#1 }
3789           { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3790           { \exp_not:o { \@currentlabel } }
3791       }
3792   }

```

`\ctex_varioref_hook:` 关于标签引用的宏包可能会修改 `\refstepcounter`。其中 `cleveref` 和 `hyperref` 宏包都会保存之前的定义, 并且它们都要求尽可能晚的被载入, 所以对我们上述的修改影响不大。需要注意的是 `varioref` 宏包, 如果它在 `CTEX` 之后被载入, 我们之前的修改将会被覆盖。较新版 `LATEX` 内核已经包含 `\labelformat`, 可以直接使用。

```

3793 \cs_new_protected:Npn \ctex_varioref_hook:
3794   {
3795     \seq_map_inline:Nn \c__ctex_headings_seq
3796       { \ctex_fix_varioref_label:n { ##1 } }
3797     \ctex_at_end_package:nn { cleveref } { \ctex_cleveref_hook: }
3798   }

```

`\ctex_fix_varioref_label:n` `varioref` 宏包的 `\labelformat` 实际上是定义一个以 `\the<#1>` 为参数的宏 `\p@<#1>`。L<sup>A</sup>T<sub>E</sub>X 在定义计数器 `<#1>` 时, 都会将 `\p@<#1>` 初始化为 `\@empty`。如果这个宏非空, 说明用户自定义了标签格式, 我们就不再修改。这里不能使用 `\exp_args:Nnc`, 因为 `c` 这种展开格式不会将参数放在花括号内。而 `\labelformat` 的定义是

```
\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

它的第二个参数必须放在花括号内, 否则将会被作为宏的定界符号。

```

3799 \cs_new_protected:Npn \ctex_fix_varioref_label:n #1
3800   {
3801     \tl_if_empty:cT { p@#1 }
3802       { \exp_args:Nne \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
3803   }

```

`patch/cleveref` `cleveref` 宏包长期未维护, 与较新的 L<sup>A</sup>T<sub>E</sub>X 内核和 `hyperref` 存在兼容问题。CT<sub>E</sub>X 提供了对 `cleveref` 的补丁, 但用户也可以选择关闭它。

当同时使用 `hyperref` 和 `cleveref` 时, `\appendix` 之后的交叉引用类型可能不正确 (例如, `appendix` 退化为 `chapter`)。这是上游 L<sup>A</sup>T<sub>E</sub>X 内核 `firstaid` 对 `cleveref` 补丁不完整所致, 与 CT<sub>E</sub>X 无关。可以使用以下方法规避:

```

\AddToHook{cmd/appendix/before}{%
  \crefalias{chapter}{appendix}%
  \crefalias{section}{subappendix}%
  \crefalias{subsection}{subsubappendix}%
  \crefalias{subsubsection}{subsubsubappendix}%
}

3804 \ctex_define:n
3805 {
3806   patch / cleveref .bool_set:N = \l__ctex_patch_cleveref_bool ,
3807   patch / cleveref .initial:n = true
3808 }

```

`\ctex_cleveref_hook:` 需要将 `cleveref` 包对应命令中 `\p@⟨counter⟩` 的参数及时展开, 以兼容 `\labelformat`。

```

\__ctex_cleveref_hook_aux:N

3809 \cs_new_protected:Npn \ctex_cleveref_hook:
3810 {
3811   \bool_if:NT \l__ctex_patch_cleveref_bool
3812   {
3813     \@ifpackageloaded { hyperref }
3814     {
3815       \@ifpackagewith { hyperref } { implicit = false }
3816       { }
3817       { \__ctex_cleveref_hook_aux:N \H@refstepcounter }
3818     }
3819     {
3820       \__ctex_cleveref_hook_aux:N \refstepcounter@noarg
3821       \__ctex_cleveref_hook_aux:N \refstepcounter@optarg
3822     }
3823     \__ctex_cleveref_hook_aux:N \appendix
3824   }
3825 }
3826 \cs_new_protected:Npn \__ctex_cleveref_hook_aux:N #1
3827 {
3828   \ctex_patch_cmd_all:NnnnTF #1
3829   {
3830     \ExplSyntaxOff
3831     \char_set_catcode_letter:n { 64 }
3832   }
3833   { \endcsname \csname the }
3834   { \expandafter \endcsname \csname the }
3835   { }
3836   { \ctex_patch_failure:N #1 }
3837 }

```

如果 `varioref` 已经被载入, 则使用它来设置。

```

3838 \cs_if_exist:NTF \labelformat
3839 { \ctex_varioref_hook: }
3840 {
3841   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3842   \RenewDocumentCommand \refstepcounter { m }
3843   {
3844     \CTEX@save@refstepcounter {#1}
3845     \CTEX@setcurrentlabel@n {#1}
3846   }
3847   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3848 }

```

### 15.8.8 载入(*scheme*) 文件

```
3849 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3850 </class|heading>
```

### 15.8.9 标题格式的 scheme 定义

下面使用 C<sub>T</sub>E<sub>X</sub> 文档类的设置方式,plain 模拟标准文档类直接定义或以 \startsection 设定的章节标题格式,chinese 汉化的标题格式。

```
3851 <*scheme&(article|book|report|beamer)>

3852 \ctex_set:nn { part }
3853 {
3854   aftertitle = \par ,
3855 <*article|book|report>
3856   hang      = false ,
3857 </article|book|report>
3858 <*plain>
3859   name      = \partname \space ,
3860 <*article|book|report>
3861   number    = \thepart ,
3862 </article|book|report>
3863 <*beamer>
3864   number    = \insertromanpartnumber ,
3865 </beamer>
3866 </plain>
3867 <*chinese>
3868   number    = \chinese { part } ,
3869 </chinese>
3870 <*article>
3871   beforeskip = 4ex ,
3872   afterskip  = 3ex ,
3873 <*plain>
3874   format     = \raggedright ,
3875   nameformat = \Large \bfseries ,
3876   aftername  = \par \nobreak ,
3877   titleformat = \huge \bfseries ,
3878   afterindent = false
3879 </plain>
3880 <*chinese>
3881   format     = \Large \bfseries \centering ,
3882   aftername  = \quad ,
3883   afterindent = true
3884 </chinese>
3885 </article>
3886 <*book|report>
3887   aftername  = \par \vskip 20 \p@ ,
3888   beforeskip = 0pt \@plus 1fil ,
3889   afterskip  = 0pt \@plus 1fil ,
3890   pagestyle  = plain ,
3891   break      = \if@openright \cleardoublepage \else \clearpage \fi ,
3892 <*plain>
3893   format     = \centering ,
3894   nameformat = \huge \bfseries ,
3895   titleformat = \Huge \bfseries
3896 </plain>
3897 <*chinese>
3898   format     = \huge \bfseries \centering
3899 </chinese>
3900 </book|report>
3901 <*beamer>
3902   format     = \centering ,
3903   nameformat = \usebeamerfont { part ~ name }
3904               \usebeamercolor [fg] { part ~ name } ,
3905   aftername  = \vskip 1em \par ,
```

```

3906     titleformat = \usebeamerfont { part ~ title }
3907 </beamer>
3908   }

3909 <*book|report>
3910 \ctex_set:nn { chapter }
3911   {
3912     pagestyle     = plain ,
3913     aftertitle    = \par ,
3914     hang          = false ,
3915     beforekip     = 50 \p@ ,
3916     afterskip     = 40 \p@ ,
3917     lofskip       = 10 \p@ ,
3918     lotskip       = 10 \p@ ,
3919     break         = \if@openright \cleardoublepage \else \clearpage \fi ,
3920 <*plain>
3921     name          = \chaptername \space ,
3922     number        = \thechapter ,
3923     format        = \raggedright ,
3924     nameformat    = \huge \bfseries ,
3925     aftername     = \par \nobreak \vskip 20 \p@ ,
3926     titleformat   = \Huge \bfseries ,
3927     afterindent   = false ,
3928     tocline       = \CTEXnumberline {#1} #2
3929 </plain>
3930 <*chinese>
3931     number        = \chinese { chapter } ,
3932     format        = \huge \bfseries \centering ,
3933     aftername     = \quad ,
3934     afterindent   = true
3935 </chinese>
3936   }
3937 </book|report>

3938 \ctex_set:nn { section }
3939   {
3940 <*article|book|report>
3941     number        = \thesection ,
3942     aftername     = \quad ,
3943     aftertitle    = \@@par ,
3944     beforekip     = 3.5ex \@plus 1ex \@minus .2ex ,
3945     afterskip     = 2.3ex \@plus .2ex ,
3946     runin         = false ,
3947     break         = \addpenalty \@secpenalty ,
3948 <*plain>
3949     format        = \Large \bfseries ,
3950     afterindent   = false
3951 </plain>
3952 <*chinese>
3953     format        = \Large \bfseries \centering ,
3954     afterindent   = true
3955 </chinese>
3956 </article|book|report>
3957 <*beamer>
3958 <*plain>
3959     name          = \sectionname \space ,
3960 </plain>
3961     format        = \centering ,
3962     number        = \insertsectionnumber ,
3963     nameformat    = \usebeamerfont { section ~ name }
3964                   \usebeamerfont [fg] { section ~ name } ,
3965     aftername     = \vskip 1em \par ,
3966     titleformat   = \usebeamerfont { section ~ title } ,
3967     aftertitle    = \par
3968 </beamer>
3969   }

3970 \ctex_set:nn { subsection }

```

```

3971 {
3972 <*article|book|report>
3973   number      = \thesubsection ,
3974   format      = \large \bfseries ,
3975   aftername   = \quad ,
3976   aftertitle  = \@@par ,
3977   beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3978   afterskip   = 1.5ex \@plus .2ex ,
3979   runin       = false ,
3980   break       = \addpenalty \@secpenalty ,
3981 <*plain>
3982   afterindent = false
3983 </plain>
3984 <*chinese>
3985   afterindent = true
3986 </chinese>
3987 </article|book|report>
3988 <*beamer>
3989 <*plain>
3990   name        = \subsectionname \space ,
3991   number      = \insertsubsectionnumber ,
3992 </plain>
3993 <*chinese>
3994   number      = \arabic { section } . \arabic { subsection } ,
3995 </chinese>
3996   format      = \centering ,
3997   nameformat  = \usebeamerfont { subsection ~ name }
3998               \usebeamercolor [fg] { subsection ~ name } ,
3999   aftername   = \vskip 1em \par ,
4000   titleformat = \usebeamerfont { subsection ~ title } ,
4001   aftertitle  = \par
4002 </beamer>
4003 }

4004 <*article|book|report>

4005 \ctex_set:nn { subsubsection }
4006 {
4007   number      = \thesubsubsection ,
4008   format      = \normalsize \bfseries ,
4009   aftername   = \quad ,
4010   aftertitle  = \@@par ,
4011   beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
4012   afterskip   = 1.5ex \@plus .2ex ,
4013   runin       = false ,
4014   break       = \addpenalty \@secpenalty ,
4015 <*plain>
4016   afterindent = false
4017 </plain>
4018 <*chinese>
4019   afterindent = true
4020 </chinese>
4021 }

4022 \ctex_set:nn { paragraph }
4023 {
4024   number      = \theparagraph ,
4025   format      = \normalsize \bfseries ,
4026   aftername   = \quad ,
4027   beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
4028   break       = \addpenalty \@secpenalty ,
4029 <*plain>
4030   afterindent = false
4031 </plain>
4032 <*chinese>
4033   afterindent = true
4034 </chinese>
4035 }

```

```

4036 \ctex_set:nn { subparagraph }
4037 {
4038   number      = \thesubparagraph ,
4039   format      = \normalsize \bfseries ,
4040   aftername   = \quad ,
4041   beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
4042   break      = \addpenalty \@secpenalty ,
4043 <*plain>
4044   afterindent = false
4045 </plain>
4046 <*chinese>
4047   afterindent = true
4048 </chinese>
4049 }

```

处理 sub3section 与 sub4section 的格式。

```

4050 \int_compare:nNnTF \g__ctex_section_depth_int > 2
4051 {
4052   \ctex_set:nn { paragraph }
4053   {
4054     aftertitle = \@@par ,
4055     afterskip  = 1ex \@plus .2ex ,
4056     runin     = false
4057   }
4058 }
4059 {
4060   \ctex_set:nn { paragraph }
4061   {
4062     afterskip = 1em ,
4063     runin    = true
4064   }
4065 }
4066 \int_compare:nNnTF \g__ctex_section_depth_int > 3
4067 {
4068   \ctex_set:nn { subparagraph }
4069   {
4070     aftertitle = \@@par ,
4071     afterskip  = 1ex \@plus .2ex ,
4072     runin     = false
4073   }
4074 }
4075 {
4076   \ctex_set:nn { subparagraph }
4077   {
4078     afterskip = 1em ,
4079     runin    = true
4080   }
4081 }
4082 \int_compare:nNnTF \g__ctex_section_depth_int > 2
4083 { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
4084 { \ctex_set:nn { subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

4085 \ctex_set:nn { appendix }
4086 <*article>
4087 { number      = \@Alph \c@section }
4088 </article>
4089 <*book|report>
4090 {
4091   name        = \appendixname \space ,
4092   number      = \@Alph \c@chapter
4093 }
4094 </book|report>
4095 </article|book|report>

```

```
4096 </scheme&(article|book|report|beamer)>
```

### 15.8.10 ctex.sty 的 heading 选项

```
4097 <*ctex|ctexheading>
```

`\c__ctex_std_class_tl` 用于记录被引入的标准文档类。

```
4098 \clist_map_inline:nn { article , book , report , beamer }
4099 {
4100   \@ifclassloaded {#1}
4101     { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
4102   { }
4103 }
```

若标准文档类被引入,则载入对应的标题定义文件。否则视 `\chapter` 是否有定义来引入 `book` 或者 `article`。

```
4104 \msg_new:nnn { ctex } { not-standard-class }
4105 {
4106   None~of~the~standard~document~classes~was~loaded.\\
4107   Heading~`#1'~is~selected.\\
4108   ctex~may~not~work~as~expected.
4109 }
<ctex> 4110 \bool_if:NTF \l__ctex_heading_bool
<ctexheading> 4111 \use:n
4112 {
4113   \tl_if_exist:NTF \c__ctex_std_class_tl
4114     { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
4115     {
4116       \cs_if_exist:NTF \chapter
4117         {
4118           \cs_if_exist:NF \if@mainmatter
4119             { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
4120           \tl_const:Nn \c__ctex_class_tl { book }
4121         }
4122         { \tl_const:Nn \c__ctex_class_tl { article } }
4123       \msg_warning:nne { ctex } { not-standard-class } { \c__ctex_class_tl }
4124     }
4125     \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }
4126   }
<ctex> 4127 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }

4128 </ctex|ctexheading>
```

### 15.8.11 标题配置文件

```
4129 <*name>
```

```
4130 \keys_set_known:nn { ctex }
4131 {
4132   contentsname = 目录 ,
4133   listfigurename = 插图 ,
4134   listtablename = 表格 ,
4135   figurename = 图 ,
4136   tablename = 表 ,
4137   abstractname = 摘要 ,
4138   indexname = 索引 ,
4139   bibname = 参考文献 ,
4140   appendixname = 附录 ,
4141   proofname = 证明 ,
4142   algorithmname = 算法 ,
4143   refname = 参考文献 ,
4144   continuation = (续) ,
4145   part / name = { 第 , 部分 } ,
```

```

4146   chapter / name = { 第 , 章 }
4147 }
4148 </name>

```

## 15.9 chinese 方案的其他设置

```
4149 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
<article|book|report> 4150 \pagestyle { headings }
```

日期格式。

```
4151 \ctex_set:n { today = small }
```

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

```

4152 \ctex_if_autoindent_touched:F
4153 { \ctex_set:n { autoindent = true } }

```

使用标题定义时的设置。首先是命题名字汉化。beamer 需要汉化定理名称。

```

4154 <!generic>
4155 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
4156 <*beamer>
4157 {
4158   \uselanguage { ChineseGBK }
4159   \languagealias { chinese } { ChineseGBK }
4160   \ctex_file_input:n { ctex-name-gbk.cfg }
4161 }
4162 {
4163   \uselanguage { ChineseUTF8 }
4164   \languagealias { chinese } { ChineseUTF8 }
4165   \ctex_file_input:n { ctex-name-utf8.cfg }
4166 }

```

让 translator 包优先查找中文翻译。

```

4167 \clist_put_left:Nn \trans@languagepath { chinese }
4168 </beamer>
4169 <*beamer>
4170 { \ctex_file_input:n { ctex-name-gbk.cfg } }
4171 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

对 beamer 以外的文档类, 若用户未设置宏包选项 zihao, 则设置 \normalsize 为五号字。beamer 不调整默认字体大小。

```

4172 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
4173 { \int_gset:Nn \g__ctex_font_size_int { 0 } }

```

对 beamer 以外的文档类, 若用户未设置宏包选项 linespread, 则设置行距初始值为  $1.3 \times 1.2 = 1.56$  倍字体大小。beamer 不调整行距。

```

4174 \fp_if_nan:nT { \l__ctex_line_spread_fp }
4175 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
4176 </!beamer>
4177 </!generic>

```

不使用标题定义时的通用设置, 注意此处 \c\_\_ctex\_std\_class\_tl 可能没有定义。

```

4178 <*generic>
4179 \tl_if_eq:NnTF \c__ctex_std_class_tl { beamer }
4180 {
4181   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
4182   {
4183     \uselanguage { ChineseGBK }
4184     \languagealias { chinese } { ChineseGBK }
4185     \ctex_file_input:n { ctex-name-gbk.cfg }

```

```

4186     }
4187     {
4188         \uselanguage { ChineseUTF8 }
4189         \languagealias { chinese } { ChineseUTF8 }
4190         \ctex_file_input:n { ctex-name-utf8.cfg }
4191     }
4192     \clist_put_left:Nn \trans@languagepath { chinese }
4193 }
4194 {
4195     \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
4196     { \ctex_file_input:n { ctex-name-gbk.cfg } }
4197     { \ctex_file_input:n { ctex-name-utf8.cfg } }
4198     \int_compare:nNnF \g__ctex_font_size_int > { -1 }
4199     { \int_gset:Nn \g__ctex_font_size_int { 0 } }
4200     \fp_if_nan:nT { \l__ctex_line_spread_fp }
4201     { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }

```

若 `ctex` 宏包与标准文档类及其衍生文档类联用, 则将载入 `indentfirst` 宏包, 实现章节标题后首个段落的段首缩进。

```

4202     \tl_if_exist:NT \c__ctex_std_class_tl
4203     { \RequirePackage { indentfirst } }
4204 }
4205 </generic>
4206 </scheme&chinese>

```

## 15.10 中文字号

```
4207 <*class|ctex|ctexsize>
```

```

\zihao 4208 \NewDocumentCommand \zihao { m }
4209   { \exp_args:Ne \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 4210 \cs_new_protected:Npn \ctex_zihao:n #1
4211   {
4212     \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
4213     { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
4214     { \msg_error:nnn { ctex } { fontsize } {#1} }
4215   }
4216 \msg_new:nnnn { ctex } { fontsize }
4217   { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
4218   {
4219     The~old~font~size~is~used~if~you~continue.\\
4220     The~available~font~sizes~are~listed~as~follow.\\
4221     \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
4222   }

```

### 15.10.1 定义中文字号

`\c__ctex_font_size_prop` 基础行距是字号的 1.2 倍, 采用 $\epsilon$ -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确<sup>20</sup>。

```

4223 \prop_new:N \l__ctex_font_size_temp_prop
4224 \seq_new:N \l__ctex_font_size_temp_seq
4225 \cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
4226   {
4227     \prop_put:Nne \l__ctex_font_size_temp_prop {#1}
4228     {
4229       { \dim_to_decimal:n {#2} }
4230       { \dim_to_decimal:n { (#2) * 6 / 5 } }
4231     }
4232     \seq_put_right:Nn \l__ctex_font_size_temp_seq {#1}
4233   }
4234 \cs_new_eq:NN \ctex_save_font_size:nn \__ctex_save_font_size:nn
4235 \group_begin:

```

```

4236 \prop_clear:N \l__ctex_font_size_temp_prop
4237 \seq_clear:N \l__ctex_font_size_temp_seq
4238 \str_case:onF { \g__ctex_font_size_system_tl }
4239 {
4240   { word }
4241   {
4242     \clist_map_inline:nn
4243     {
4244       { 8 } { 5 bp } ,
4245       { 7 } { 5.5 bp } ,
4246       { -6 } { 6.5 bp } ,
4247       { 6 } { 7.5 bp } ,
4248       { -5 } { 9 bp } ,
4249       { 5 } { 10.5 bp } ,
4250       { -4 } { 12 bp } ,
4251       { 4 } { 14 bp } ,
4252       { -3 } { 15 bp } ,
4253       { 3 } { 16 bp } ,
4254       { -2 } { 18 bp } ,
4255       { 2 } { 22 bp } ,
4256       { -1 } { 24 bp } ,
4257       { 1 } { 26 bp } ,
4258       { -0 } { 36 bp } ,
4259       { 0 } { 42 bp }
4260     }
4261     { \__ctex_save_font_size:nn #1 }
4262   }
4263   { letterpress }
4264   {
4265     \clist_map_inline:nn
4266     {
4267       { 8 } { 4 bp } ,
4268       { 7 } { 5.25 bp } ,
4269       { -6 } { 7.5 bp } ,
4270       { 6 } { 8 bp } ,
4271       { -5 } { 9 bp } ,
4272       { 5 } { 10.5 bp } ,
4273       { -4 } { 12 bp } ,
4274       { 4 } { 14 bp } ,
4275       { -3 } { 15 bp } ,
4276       { 3 } { 16 bp } ,
4277       { -2 } { 18 bp } ,
4278       { 2 } { 21 bp } ,
4279       { -1 } { 24 bp } ,
4280       { 1 } { 28 bp } ,
4281       { -0 } { 36 bp } ,
4282       { 0 } { 42 bp }
4283     }
4284     { \__ctex_save_font_size:nn #1 }
4285   }
4286 }
4287 {
4288   \file_if_exist:nTF { ctex-fontsize- \g__ctex_font_size_system_tl .def }
4289   {
4290     \ctex_file_input:n
4291     { ctex-fontsize- \g__ctex_font_size_system_tl .def }
4292   }
4293   {
4294     \msg_error:nne { ctex } { fontsize-system-not-found }
4295     { \g__ctex_font_size_system_tl }
4296     \clist_map_inline:nn
4297     {
4298       { 8 } { 5 bp } ,
4299       { 7 } { 5.5 bp } ,
4300       { -6 } { 6.5 bp } ,
4301       { 6 } { 7.5 bp } ,
4302       { -5 } { 9 bp } ,

```

```

4303         { 5 } { 10.5 bp } ,
4304         { -4 } { 12 bp } ,
4305         { 4 } { 14 bp } ,
4306         { -3 } { 15 bp } ,
4307         { 3 } { 16 bp } ,
4308         { -2 } { 18 bp } ,
4309         { 2 } { 22 bp } ,
4310         { -1 } { 24 bp } ,
4311         { 1 } { 26 bp } ,
4312         { -0 } { 36 bp } ,
4313         { 0 } { 42 bp }
4314     }
4315     { \__ctex_save_font_size:nn #1 }
4316 }
4317 }
4318 \exp_args:NNe \prop_const_from_keyval:Nn \c__ctex_font_size_prop
4319 { \prop_to_keyval:N \l__ctex_font_size_temp_prop }
4320 \exp_args:NNe \seq_const_from_clist:Nn \c__ctex_font_size_seq
4321 { \seq_use:Nn \l__ctex_font_size_temp_seq { , } }
4322 \group_end:
4323 \msg_new:nnnn { ctex } { fontsize-system-not-found }
4324 {
4325     CTeX~font~size~system~`#1'~could~not~be~found.
4326     \\ The~`word'~system~will~be~used~instead.
4327 }
4328 { You~may~need~to~create~`ctex-fontsize-#1.def'. }

\ctex_declare_math_sizes:nnnn 4329 \cs_new_protected:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
4330 {
4331     \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
4332     \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
4333 }

\__ctex_get_font_sizes:Nn 4334 \cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
4335 {
4336     \tl_clear:N #1
4337     \tl_map_inline:nn {#2}
4338     {
4339         \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
4340         { \tl_put_right:Ne #1 { { \tl_head:N \l__ctex_tmp_tl } } }
4341         { \tl_put_right:Ne #1 { { \dim_to_decimal:n { ##1 } } } }
4342     }
4343 }

4344 \clist_map_inline:nn
4345 {
4346     { 8 } { 8 } { 5pt } { 5pt } ,
4347     { 7 } { 7 } { 5pt } { 5pt } ,
4348     { -6 } { -6 } { 5pt } { 5pt } ,
4349     { 6 } { 6 } { 5pt } { 5pt } ,
4350     { -5 } { -5 } { 6pt } { 5pt } ,
4351     { 5 } { 5 } { 7pt } { 5pt } ,
4352     { -4 } { -4 } { 8pt } { 6pt } ,
4353     { 4 } { 4 } { 5 } { 6 } ,
4354     { -3 } { -3 } { -4 } { -5 } ,
4355     { 3 } { 3 } { 4 } { 5 } ,
4356     { -2 } { -2 } { -3 } { -4 } ,
4357     { 2 } { 2 } { 3 } { 4 } ,
4358     { -1 } { -1 } { -2 } { -3 } ,
4359     { 1 } { 1 } { 2 } { 3 } ,
4360     { -0 } { -0 } { -1 } { -2 } ,
4361     { 0 } { 0 } { 1 } { 2 }
4362 }
4363 { \ctex_declare_math_sizes:nnnn #1 }

```

## 15.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 4364 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
4365 {
4366   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
4367   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
4368   { \msg_error:nnn { ctex } { fontsize } {#2} }
4369 }
4370 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
4371 { \cs_set_protected:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

4372 \int_case:nn { \g__ctex_font_size_int }
4373 {
4374   { 0 } { \file_input:n { ctex-c5size.clo } }
4375   { 1 } { \file_input:n { ctex-cs4size.clo } }
4376 }

4377 </class>ctex|ctexsize>
4378 <*c5size>
4379 \ctex_set_font_size:Nnn \normalsize { 5 }
4380 {
4381   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
4382   \abovedisplayshortskip \z@ \@plus3\p@
4383   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
4384   \belowdisplayskip \abovedisplayskip
4385   \let\@listi\@listI
4386 }
4387 \ctex_set_font_size:Nnn \small { -5 }
4388 {
4389   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
4390   \abovedisplayshortskip \z@ \@plus2\p@
4391   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
4392   \def\@listi{\leftmargin\leftmargini
4393     \topsep 4\p@ \@plus2\p@ \@minus2\p@
4394     \parsep 2\p@ \@plus\p@ \@minus\p@
4395     \itemsep \parsep}
4396   \belowdisplayskip \abovedisplayskip
4397 }
4398 \ctex_set_font_size:Nnn \footnotesize { 6 }
4399 {
4400   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
4401   \abovedisplayshortskip \z@ \@plus\p@
4402   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
4403   \def\@listi{\leftmargin\leftmargini
4404     \topsep 3\p@ \@plus\p@ \@minus\p@
4405     \parsep 2\p@ \@plus\p@ \@minus\p@
4406     \itemsep \parsep}
4407   \belowdisplayskip \abovedisplayskip
4408 }
4409 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
4410 \ctex_set_font_size:Nnn \tiny { 7 } { }
4411 \ctex_set_font_size:Nnn \large { -4 } { }
4412 \ctex_set_font_size:Nnn \Large { -3 } { }
4413 \ctex_set_font_size:Nnn \LARGE { -2 } { }
4414 \ctex_set_font_size:Nnn \huge { 2 } { }
4415 \ctex_set_font_size:Nnn \Huge { 1 } { }
4416 </c5size>
4417 <*cs4size>
4418 \ctex_set_font_size:Nnn \normalsize { -4 }
4419 {
4420   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
4421   \abovedisplayshortskip \z@ \@plus3\p@
4422   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
4423   \belowdisplayskip \abovedisplayskip
4424   \let\@listi\@listI
4425 }
4426 \ctex_set_font_size:Nnn \small { 5 }
4427 {

```

```

4428 \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
4429 \abovedisplayshortskip \z@ \@plus3\p@
4430 \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
4431 \def\@listi{\leftmargin\leftmargini
4432 \topsep 9\p@ \@plus3\p@ \@minus5\p@
4433 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
4434 \itemsep \parsep}
4435 \belowdisplayskip \abovedisplayskip
4436 }
4437 \ctex_set_font_size:Nnn \footnotesize { -5 }
4438 {
4439 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
4440 \abovedisplayshortskip \z@ \@plus3\p@
4441 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
4442 \def\@listi{\leftmargin\leftmargini
4443 \topsep 6\p@ \@plus2\p@ \@minus2\p@
4444 \parsep 3\p@ \@plus2\p@ \@minus\p@
4445 \itemsep \parsep}
4446 \belowdisplayskip \abovedisplayskip
4447 }
4448 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
4449 \ctex_set_font_size:Nnn \tiny { -6 } { }
4450 \ctex_set_font_size:Nnn \large { -3 } { }
4451 \ctex_set_font_size:Nnn \Large { -2 } { }
4452 \ctex_set_font_size:Nnn \LARGE { 2 } { }
4453 \ctex_set_font_size:Nnn \huge { -1 } { }
4454 \ctex_set_font_size:Nnn \Huge { 1 } { }
4455 </cs4size>
<ctexsize> 4456 \normalsize
4457 <*class|ctex>

```

## 15.11 更新行距

`\l__ctex_line_spread_fp` 被设置了才有必要更新行距和 `\footnotesep`。

```

4458 \fp_if_nan:nF { \l__ctex_line_spread_fp }
4459 {
4460 \exp_args:Ne \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距，可能导致脚注的间距与行距不协调，需要调整 `\footnotesep`。标准文档类对 `\footnotesep` 的设置是，字体大小为 `\footnotesize` 时 `\strutbox` 的高度（默认值是 `.7\baselineskip`）。我们沿用这个设置方法，只需要更新具体的大小。

```

4461 \group_begin: \footnotesize \exp_args:NNNo \group_end:
4462 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
4463 }

```

激活默认字体大小，更新行距、`\parindent` 和 `\CJKglue`。

```
4464 \normalsize
```

## 15.12 其他功能

`\CTeX` `ctex-faq.sty` 中的定义是

```
\DeclareRobustCommand\CTeX{\mathbb{C}}\kern-.05em\TeX}
```

然而 `\mathbb` 未必有定义，这里就不采用它了，只定义最简单的形式。`CTeX` 可以直接用在 PDF 书签中。

```
4465 \NewDocumentCommand \CTeX { }
```

```

4466 { C \TeX }
4467 \ctex_at_end_package:nn { hyperref }
4468 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }

```

`captiondelimitter` 过时选项。

```

4469 \ctex_define:n
4470 {
4471   captiondelimitter .code:n =
4472   {
4473     \ctex_deprecated_option:n
4474     { You~can~load~the~package~`caption'~to~get~its~functionality. }
4475   }
4476 }
4477 </class|ctex>

```

### 15.12.1 列表环境的缩进

`\verse` 只在使用文档类的时候修改诗歌和引用环境的缩进。  
`\quotation`

```

4478 <*scheme&chinese&(article|book|report)>
4479 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
4480 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
4481 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
4482 </scheme&chinese&(article|book|report)>
4483 <*class|ctex>

```

### 15.13 载入中文字库

`\ctex_fontset_error:n` 字库不可用时给出紧急错误信息, 停止读取定义文件。

```

4484 \cs_new_protected:Npn \ctex_fontset_error:n #1
4485 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
4486 \msg_new:nnn { ctex } { fontset-unavailable }
4487 { CTeX`fontset~`#1'~is~unavailable~in~current~mode. }

```

`\ctex_fontset_case:nnn` 3 个参数依次为 pdf<sub>TeX</sub>、up<sub>TeX</sub> 和 X<sub>Y</sub>TeX/Lua<sub>TeX</sub>。

```

4488 \cs_new:Npx \ctex_fontset_case:nnn #1#2#3
4489 {
4490   \sys_if_engine_pdftex:TF
4491   {#1}
4492   { \sys_if_engine_uptex:TF {#2} {#3} }
4493 }

```

`\ctex_fontset_case:nnnn` 4 个参数依次为 pdf<sub>TeX</sub>(生成 PDF)、pdf<sub>TeX</sub>(生成 DVI)、up<sub>TeX</sub> 和 X<sub>Y</sub>TeX/Lua<sub>TeX</sub>。

```

4494 \cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
4495 {
4496   \sys_if_engine_pdftex:TF
4497   { \sys_if_output_pdf:TF {#1} {#2} }
4498   { \sys_if_engine_uptex:TF {#3} {#4} }
4499 }

```

`\ctex_detect_platform:` 根据操作系统判断默认字体配置。

```

4500 \cs_new_protected:Npn \ctex_detect_platform:
4501 {
4502   \sys_if_platform_windows:TF

```

```

4503     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
4504     {
4505       \ctex_if_platform_macos:TF
4506         { \tl_gset:Nn \g__ctex_fontset_tl { mac } }
4507         { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
4508     }
4509 }

```

`\ctex_if_platform_macos:TF` 以特定字体判断 macOS 系统。

```

4510 \cs_new_protected:Npn \ctex_if_platform_macos:TF
4511 { \file_if_exist:nTF { /System/Library/Fonts/Menlo.ttc } }

```

`\ctex_load_fontset:` 如果用户没有指定字体,则探测操作系统,载入相应的字体配置。

```

4512 \cs_new_protected:Npn \ctex_load_fontset:
4513 {
4514   \tl_if_empty:NTF \g__ctex_fontset_tl
4515     { \ctex_detect_platform: }
4516     {
4517       \bool_lazy_or:nnTF
4518         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
4519         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
4520         {
4521           \msg_warning:nnee { ctex } { deprecated-fontset }
4522           { \g__ctex_fontset_tl } { windows }
4523           \tl_gset:Nn \g__ctex_fontset_tl { windows }
4524         }
4525       {
4526         \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
4527         {
4528           \use:e
4529             {
4530               \ctex_detect_platform:
4531               \msg_error:nnee { ctex } { fontset-not-found }
4532               { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
4533             }
4534         }
4535       }
4536     }
4537   \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
4538 }
4539 \msg_new:nnn { ctex } { deprecated-fontset }
4540 { CTeX~fontset~#1~is~deprecated.\\ Fontset~#2~will~be~used~instead. }
4541 \msg_new:nnnn { ctex } { fontset-not-found }
4542 { CTeX~fontset~#1~could~not~be~found.\\ Fontset~#2~will~be~used~instead. }
4543 { You~may~run~mktexlsr~firstly. }
4544 \@onlypreamble \ctex_load_fontset:

```

**fontset** 在导言区通过 `\ctexset` 载入中文字库的选项。

```

4545 \ctex_define:n
4546 {
4547   fontset .code:n =
4548   {
4549     \ctex_if_preamble:TF
4550     {
4551       \str_if_eq:eeTF {#1} { none }
4552       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4553       {
4554         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4555         {
4556           \tl_gset:Ne \g__ctex_fontset_tl {#1}
4557           \ctex_load_fontset:
4558         }
4559       }
4560     }
4561   }

```

```

4560             \msg_error:nnee { ctex } { fontset-loaded }
4561             { \g__ctex_fontset_tl } {#1}
4562         }
4563     }
4564 }
4565 { \msg_error:nn { ctex } { fontset-only-preamble } }
4566 }
4567 }
4568 \msg_new:nnnn { ctex } { fontset-loaded }
4569 {
4570     CTeX~fontset~`#1'~has~been~loaded.
4571     \str_if_eq:nnF {#1} {#2} { \ \ Fontset~`#2'~will~be~ignored. }
4572 }
4573 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4574 \msg_new:nnn { ctex } { fontset-only-preamble }
4575 { The~`fontset'~option~can~be~used~only~in~preamble. }

```

载入中文字库。

```

4576 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4577 { \ctex_load_fontset: }

```

## 15.14 宏包配置文件

### 15.14.1 ctex.cfg

```

4578 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4579 </class|ctex>
4580 <*config>
4581 %%
4582 </config>

```

### 15.14.2 ctexopts.cfg

这里仅为配置文件示例:使用 Windows 下的字体设置。

```

4583 <*ctexopts>
4584 %%
4585 %% \ctex_set:nn { option } { fontset = windows }
4586 </ctexopts>

```

## 15.15 字体定义文件

### 15.15.1 传统定义方式

```

4587 <*c19|c70>
4588 %%
4589 %% Chinese characters
4590 %%
<c19> 4591 %% character set: GBK (extension of GB 2312)
<c70> 4592 %% character set: Unicode
4593 %% font encoding: Unicode
4594 %%
4595 </c19|c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 4596 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 4597 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 4598 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 4599 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 4600 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}

```

```
<tt&c70> 4601 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}
```

up $\LaTeX$  使用的字体族。up $\LaTeX$  在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```
<rm&jy2> 4602 \DeclareKanjiFamily{JY2}{zhrm}{}  
<rm&jt2> 4603 \DeclareKanjiFamily{JT2}{zhrm}{}  
<sf&jy2> 4604 \DeclareKanjiFamily{JY2}{zhshf}{}  
<sf&jt2> 4605 \DeclareKanjiFamily{JT2}{zhshf}{}  
<tt&jy2> 4606 \DeclareKanjiFamily{JY2}{zhhtt}{}  
<tt&jt2> 4607 \DeclareKanjiFamily{JT2}{zhhtt}{}
```

在 up $\LaTeX$  字体族 zhrm 下采用如下 Fallback 机制:形状 s1 优先保留加粗字重,形状 it 优先保留楷书字形。

```
4608 <*rm>  
4609 <*c19>  
4610 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}  
4611 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}  
4612 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}  
4613 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}  
4614 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}  
4615 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}  
4616 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}  
4617 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}  
4618 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}  
4619 </c19>  
4620 <*c70>  
4621 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}  
4622 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}  
4623 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}  
4624 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}  
4625 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}  
4626 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}  
4627 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}  
4628 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}  
4629 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}  
4630 </c70>  
4631 <*jy2>  
4632 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{}  
4633 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{}  
4634 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{}  
4635 \DeclareFontShape{JY2}{zhrm}{m}{sl}{<-> upzhserifit-h}{}  
4636 \DeclareFontShape{JY2}{zhrm}{b}{sl}{<-> upzhserifb-h}{}  
4637 \DeclareFontShape{JY2}{zhrm}{bx}{sl}{<-> upzhserifb-h}{}  
4638 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{}  
4639 \DeclareFontShape{JY2}{zhrm}{b}{it}{<-> upzhserifit-h}{}  
4640 \DeclareFontShape{JY2}{zhrm}{bx}{it}{<-> upzhserifit-h}{}  
4641 </jy2>  
4642 <*jt2>  
4643 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{}  
4644 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{}  
4645 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{}  
4646 \DeclareFontShape{JT2}{zhrm}{m}{sl}{<-> upzhserifit-v}{}  
4647 \DeclareFontShape{JT2}{zhrm}{b}{sl}{<-> upzhserifb-v}{}  
4648 \DeclareFontShape{JT2}{zhrm}{bx}{sl}{<-> upzhserifb-v}{}  
4649 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{}  
4650 \DeclareFontShape{JT2}{zhrm}{b}{it}{<-> upzhserifit-v}{}  
4651 \DeclareFontShape{JT2}{zhrm}{bx}{it}{<-> upzhserifit-v}{}  
4652 </jt2>  
4653 </rm>  
  
4654 <*sf>  
4655 <*c19>  
4656 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}  
4657 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}  
4658 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}  
4659 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
```

```

4660 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4661 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4662 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4663 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4664 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4665 </c19>
4666 <*c70>
4667 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4668 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4669 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4670 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4671 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4672 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4673 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4674 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4675 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4676 </c70>
4677 <*jy2>
4678 \DeclareFontShape{JY2}{zhsf}{m}{n}{<-> upzhsans-h}{}
4679 \DeclareFontShape{JY2}{zhsf}{b}{n}{<-> upzhsansb-h}{}
4680 \DeclareFontShape{JY2}{zhsf}{bx}{n}{<-> upzhsansb-h}{}
4681 \DeclareFontShape{JY2}{zhsf}{m}{sl}{<-> upzhsans-h}{}
4682 \DeclareFontShape{JY2}{zhsf}{b}{sl}{<-> upzhsansb-h}{}
4683 \DeclareFontShape{JY2}{zhsf}{bx}{sl}{<-> upzhsansb-h}{}
4684 \DeclareFontShape{JY2}{zhsf}{m}{it}{<-> upzhsans-h}{}
4685 \DeclareFontShape{JY2}{zhsf}{b}{it}{<-> upzhsansb-h}{}
4686 \DeclareFontShape{JY2}{zhsf}{bx}{it}{<-> upzhsansb-h}{}
4687 </jy2>
4688 <*jt2>
4689 \DeclareFontShape{JT2}{zhsf}{m}{n}{<-> upzhsans-v}{}
4690 \DeclareFontShape{JT2}{zhsf}{b}{n}{<-> upzhsansb-v}{}
4691 \DeclareFontShape{JT2}{zhsf}{bx}{n}{<-> upzhsansb-v}{}
4692 \DeclareFontShape{JT2}{zhsf}{m}{sl}{<-> upzhsans-v}{}
4693 \DeclareFontShape{JT2}{zhsf}{b}{sl}{<-> upzhsansb-v}{}
4694 \DeclareFontShape{JT2}{zhsf}{bx}{sl}{<-> upzhsansb-v}{}
4695 \DeclareFontShape{JT2}{zhsf}{m}{it}{<-> upzhsans-v}{}
4696 \DeclareFontShape{JT2}{zhsf}{b}{it}{<-> upzhsansb-v}{}
4697 \DeclareFontShape{JT2}{zhsf}{bx}{it}{<-> upzhsansb-v}{}
4698 </jt2>
4699 </sf>

4700 <*tt>
4701 <*c19>
4702 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4703 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4704 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4705 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4706 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4707 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4708 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4709 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4710 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4711 </c19>
4712 <*c70>
4713 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4714 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4715 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4716 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4717 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4718 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4719 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4720 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4721 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4722 </c70>
4723 <*jy2>
4724 \DeclareFontShape{JY2}{zh tt}{m}{n}{<-> upzhmono-h}{}
4725 \DeclareFontShape{JY2}{zh tt}{b}{n}{<-> upzhmono-h}{}
4726 \DeclareFontShape{JY2}{zh tt}{bx}{n}{<-> upzhmono-h}{}

```

```

4727 \DeclareFontShape{JY2}{zh tt}{m}{sl}{<-> upzhmono-h}{ }
4728 \DeclareFontShape{JY2}{zh tt}{b}{sl}{<-> upzhmono-h}{ }
4729 \DeclareFontShape{JY2}{zh tt}{bx}{sl}{<-> upzhmono-h}{ }
4730 \DeclareFontShape{JY2}{zh tt}{m}{it}{<-> upzhmono-h}{ }
4731 \DeclareFontShape{JY2}{zh tt}{b}{it}{<-> upzhmono-h}{ }
4732 \DeclareFontShape{JY2}{zh tt}{bx}{it}{<-> upzhmono-h}{ }
4733 </jy2>
4734 <*jt2>
4735 \DeclareFontShape{JT2}{zh tt}{m}{n}{<-> upzhmono-v}{ }
4736 \DeclareFontShape{JT2}{zh tt}{b}{n}{<-> upzhmono-v}{ }
4737 \DeclareFontShape{JT2}{zh tt}{bx}{n}{<-> upzhmono-v}{ }
4738 \DeclareFontShape{JT2}{zh tt}{m}{sl}{<-> upzhmono-v}{ }
4739 \DeclareFontShape{JT2}{zh tt}{b}{sl}{<-> upzhmono-v}{ }
4740 \DeclareFontShape{JT2}{zh tt}{bx}{sl}{<-> upzhmono-v}{ }
4741 \DeclareFontShape{JT2}{zh tt}{m}{it}{<-> upzhmono-v}{ }
4742 \DeclareFontShape{JT2}{zh tt}{b}{it}{<-> upzhmono-v}{ }
4743 \DeclareFontShape{JT2}{zh tt}{bx}{it}{<-> upzhmono-v}{ }
4744 </jt2>
4745 </tt>

```

## 15.15.2 预定义字库

```
4746 <*fontset>
```

### 15.15.2.1 adobe

\pdfmapline 不支持 OpenType 字体，因而 adobe 字体集在 pdf 模式下就没有定义。fandol 的情况类似。

```

4747 <*adobe>
4748 \ctex_fontset_case:n n n n
4749 { \ctex_fontset_error:n { adobe } }
4750 {
4751   \ctex_zhmap_case:n n n
4752   {
4753     \setCJKmainfont { AdobeSongStd-Light.otf }
4754     [
4755       cmap          = UniGB-UTF16-H,
4756       BoldFont     = AdobeHeitiStd-Regular.otf,
4757       ItalicFont   = AdobeKaitiStd-Regular.otf
4758     ]
4759     \setCJKsansfont { AdobeHeitiStd-Regular.otf }
4760     [ cmap = UniGB-UTF16-H ]
4761     \setCJKmonofont { AdobeFangsongStd-Regular.otf }
4762     [ cmap = UniGB-UTF16-H ]
4763     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf }
4764     [ cmap = UniGB-UTF16-H ]
4765     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular.otf }
4766     [ cmap = UniGB-UTF16-H ]
4767     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular.otf }
4768     [ cmap = UniGB-UTF16-H ]
4769     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular.otf }
4770     [ cmap = UniGB-UTF16-H ]
4771     \ctex_punct_set:n { adobe }
4772     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4773     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4774     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4775     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4776     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4777   }
4778   {
4779     \ctex_load_zhmap:n n n n { rm } { zhhei } { zhfs } { adobe }
4780     \ctex_punct_set:n { adobe }
4781     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4782     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4783     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4784   }

```

```

4785     { \ctex_fontset_error:n { adobe } }
4786   }
4787   {
4788     \ctex_set_upfonts:nnnnnn
4789     { AdobeSongStd-Light.otf      }
4790     { AdobeHeitiStd-Regular.otf   }
4791     { AdobeKaitiStd-Regular.otf   }
4792     { AdobeHeitiStd-Regular.otf   }
4793     { AdobeHeitiStd-Regular.otf   }
4794     { AdobeFangsongStd-Regular.otf }
4795     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4796     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4797     \ctex_set_upfamily:nnn { zhfs  } { upzhmono  } {}
4798     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4799   }
4800   {
4801     \setCJKmainfont { AdobeSongStd-Light      }
4802     [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
4803     \setCJKsansfont { AdobeHeitiStd-Regular   }
4804     \setCJKmonofont { AdobeFangsongStd-Regular }
4805     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light      }
4806     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular   }
4807     \setCJKfamilyfont { zhfs  } { AdobeFangsongStd-Regular }
4808     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular   }
4809   }
4810 </adobe>

```

### 15.15.2.2 fandol

```

4811 <*fandol>
4812 \ctex_fontset_case:nnnn
4813 { \ctex_fontset_error:n { fandol } }
4814 {
4815   \ctex_zhmap_case:nnn
4816   {
4817     \setCJKmainfont { FandolSong-Regular.otf }
4818     [
4819       cmap      = UniGB-UTF16-H,
4820       BoldFont  = FandolSong-Bold.otf,
4821       ItalicFont = FandolKai-Regular.otf
4822     ]
4823     \setCJKsansfont { FandolHei-Regular.otf }
4824     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4825     \setCJKmonofont { FandolFang-Regular.otf }
4826     [ cmap = UniGB-UTF16-H ]
4827     \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
4828     [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
4829     \setCJKfamilyfont { zhhei } { FandolHei-Regular.otf }
4830     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4831     \setCJKfamilyfont { zhfs  } { FandolFang-Regular.otf }
4832     [ cmap = UniGB-UTF16-H ]
4833     \setCJKfamilyfont { zhkai } { FandolKai-Regular.otf }
4834     [ cmap = UniGB-UTF16-H ]
4835     \ctex_punct_set:n { fandol }
4836     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4837     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4838     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs  }
4839     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4840     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4841     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4842   }
4843   {
4844     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { fandol }
4845     \ctex_punct_set:n { fandol }
4846     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4847     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }

```

```

4848     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4849   }
4850   { \ctex_fontset_error:n { fandol } }
4851 }
4852 {
4853   \ctex_set_upfonts:nnnnnn
4854   { FandolSong-Regular.otf }
4855   { FandolSong-Bold.otf }
4856   { FandolKai-Regular.otf }
4857   { FandolHei-Regular.otf }
4858   { FandolHei-Bold.otf }
4859   { FandolFang-Regular.otf }
4860   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4861   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4862   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4863   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4864 }
4865 {
4866   \setCJKmainfont { FandolSong-Regular }
4867   [
4868     Extension = .otf,
4869     BoldFont = FandolSong-Bold,
4870     ItalicFont = FandolKai-Regular
4871   ]
4872   \setCJKsansfont { FandolHei-Regular }
4873   [ Extension = .otf, BoldFont = FandolHei-Bold ]
4874   \setCJKmonofont { FandolFang-Regular }
4875   [ Extension = .otf ]
4876   \setCJKfamilyfont { zhsong } { FandolSong-Regular }
4877   [ Extension = .otf, BoldFont = FandolSong-Bold ]
4878   \setCJKfamilyfont { zhhei } { FandolHei-Regular }
4879   [ Extension = .otf, BoldFont = FandolHei-Bold ]
4880   \setCJKfamilyfont { zhfs } { FandolFang-Regular }
4881   [ Extension = .otf ]
4882   \setCJKfamilyfont { zhkai } { FandolKai-Regular }
4883   [ Extension = .otf ]
4884 }
4885 </fandol>

```

### 15.15.2.3 founder

```

4886 <*founder>
4887 \ctex_fontset_case:nnn
4888 {
4889   \ctex_zhmap_case:nnn
4890   {
4891     \setCJKmainfont { FZSSK.TTF }
4892     [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
4893     \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
4894     \setCJKmonofont { FZFSK.TTF }
4895     \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
4896     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4897     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4898     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4899     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4900     \setCJKfamilyfont { zhyou } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
4901     \ctex_punct_set:n { founder }
4902     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4903     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4904     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4905     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4906     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4907     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4908     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4909   }
4910   {
4911     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { founder }

```

```

4912     \ctex_punct_set:n { founder }
4913     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4914     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4915     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4916   }
4917   { \ctex_fontset_error:n { founder } }
4918 }
4919 {
4920   \ctex_set_upfonts:nnnnnn
4921   { FZSSK.TTF }
4922   { FZXBSK.TTF }
4923   { FZKTK.TTF }
4924   { FZXH1K.TTF }
4925   { FZHTK.TTF }
4926   { FZFSK.TTF }
4927   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4928   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4929   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4930   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4931   \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4932   \ctex_set_upfamily:nnn { zhyou } { upscht } {}
4933   \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4934   \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4935 }
4936 {
4937   \setCJKmainfont { FZShuSong-Z01 }
4938   [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
4939   \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
4940   \setCJKmonofont { FZFangSong-Z02 }
4941   \setCJKfamilyfont { zhsong } { FZShuSong-Z01 }
4942   [ BoldFont = FZXiaoBiaoSong-B05 ]
4943   \setCJKfamilyfont { zhhei } { FZHei-B01 }
4944   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4945   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4946   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4947   \setCJKfamilyfont { zhyou } { FZXiYuan-M01 }
4948   [ BoldFont = FZZhunYuan-M02 ]
4949 }
4950 </founder>

```

#### 15.15.2.4 hanyi

```

4951 <*hanyi>
4952 \ctex_fontset_case:nnn
4953 {
4954   \ctex_zhmap_case:nnn
4955   {
4956     \setCJKmainfont { HYShuSongErS.ttf }
4957     [ BoldFont = HYZhongSongS.ttf, ItalicFont = HYKaiTiS.ttf ]
4958     \setCJKsansfont { HYZhongHeiS.ttf } [ BoldFont = HYDaHeiS.ttf ]
4959     \setCJKmonofont { HYFangSongS.ttf }
4960     \setCJKfamilyfont { zhsong } { HYShuSongErS.ttf }
4961     [ BoldFont = HYZhongSongS.ttf ]
4962     \setCJKfamilyfont { zhhei } { HYZhongHeiS.ttf }
4963     [ BoldFont = HYDaHeiS.ttf ]
4964     \setCJKfamilyfont { zhkai } { HYKaiTiS.ttf }
4965     \setCJKfamilyfont { zhfs } { HYFangSongS.ttf }
4966     \ctex_punct_set:n { hanyi }
4967     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4968     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4969     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4970     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4971     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4972     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4973   }
4974   {
4975     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { hanyi }

```

```

4976     \ctex_punct_set:n { hanyi }
4977     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4978     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4979     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4980   }
4981   { \ctex_fontset_error:n { hanyi } }
4982 }
4983 {
4984   \ctex_set_upfonts:nnnnnn
4985   { HYShuSongErS.ttf }
4986   { HYZhongSongS.ttf }
4987   { HYKaiTiS.ttf }
4988   { HYZhongHeiS.ttf }
4989   { HYDaHeiS.ttf }
4990   { HYFangSongS.ttf }
4991   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4992   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4993   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4994   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4995 }
4996 {
4997   \setCJKmainfont { HYShuSongEr~S }
4998   [ BoldFont = HYZhongSong~S, ItalicFont = HYKaiTi~S ]
4999   \setCJKsansfont { HYZhongHei~S } [ BoldFont = HYDaHei~S ]
5000   \setCJKmonofont { HYFangSong~S }
5001   \setCJKfamilyfont { zhsong } { HYShuSongEr~S }
5002   [ BoldFont = HYZhongSong~S ]
5003   \setCJKfamilyfont { zhhei } { HYZhongHei~S }
5004   [ BoldFont = HYDaHei~S ]
5005   \setCJKfamilyfont { zhkai } { HYKaiTi~S }
5006   \setCJKfamilyfont { zhfs } { HYFangSong~S }
5007 }
5008 </hanyi>

```

### 15.15.2.5 mac 相关

按 [Issue 351](#) 的讨论,以 El Capitan 为分界,分别设置 `macold`(El Capitan 之前)和 `macnew`(El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。从 macOS 15 (Sequoia) 开始,苹方字体转为 `downloadable`,不再位于传统路径,故增加版本号检测作为后备。

```

5009 <*mac>
5010 \tl_new:N \g__ctex_macos_ver_tl
5011 \msg_new:nnn { ctex } { macos-version-detect-failed }
5012 { Cannot~detect~macOS~version.~Falling~back~to~macold~fontset. }

```

为方便人类阅读,把 Lua $\TeX$  分支挪到后面,在不忽略空格的 `catcode scheme` 下编写。这样 Lua 代码就不必以 `~` 代替空格。非 Lua $\TeX$  分支,在末尾使用 `\use_none_delimit_by_q_stop:w` 来跳过 Lua $\TeX$  分支。

```

5013 \sys_if_engine luatex:F
5014 {
5015   \ior_new:N \g__ctex_macos_ver_ior
5016   \bool_new:N \g__ctex_macos_ver_found_bool
5017   \file_if_exist:nT { /System/Library/CoreServices/SystemVersion.plist }
5018   {
5019     \ior_open:Nn \g__ctex_macos_ver_ior
5020     { /System/Library/CoreServices/SystemVersion.plist }
5021     \ior_map_inline:Nn \g__ctex_macos_ver_ior
5022     {
5023       \bool_if:NTF \g__ctex_macos_ver_found_bool
5024       {
5025         \tl_set:Nn \l_tmpa_tl {#1}
5026         \regex_replace_once:nnN
5027         { .*? <string> (\d+) .* } { \1 } \l_tmpa_tl
5028         \tl_gset:NV \g__ctex_macos_ver_tl \l_tmpa_tl

```

```

5029         \ior_map_break:
5030     }
5031     {
5032         \tl_if_in:nnT {#1} { ProductVersion }
5033         { \bool_gset_true:N \g__ctex_macos_ver_found_bool }
5034     }
5035 }
5036 \ior_close:N \g__ctex_macos_ver_ior
5037 }
5038 \use_none_delimit_by_q_stop:w
5039 }
5040 \char_set_catcode_space:n { 32 }
5041 \lua_now:n
5042 {
5043     local f = io.open("/System/Library/CoreServices/SystemVersion.plist", "r");
5044     if f then
5045         local content = f:read("*a");
5046         f:close();
5047         local ver = content:match("ProductVersion.-<string>(.)</string>");
5048         if ver then
5049             local dot = ver:find(".", 1, true);
5050             local major = ver:sub(1, dot - 1) or ver;
5051             if major then
5052                 token.set_macro("g__ctex_macos_ver_tl", major, "global");
5053             end;
5054         end;
5055     end
5056 }
5057 \char_set_catcode_ignore:n { 32 }
5058 \use_none_delimit_by_q_stop:w \q_stop
5059 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
5060 { \ctex_file_input:n { ctex-fontset-macnew.def } }
5061 {
5062     \tl_if_empty:NTF \g__ctex_macos_ver_tl
5063     {
5064         \msg_warning:nn { ctex } { macos-version-detect-failed }
5065         \ctex_file_input:n { ctex-fontset-macold.def }
5066     }
5067     {
5068         \int_compare:nNnTF { \g__ctex_macos_ver_tl } < { 15 }
5069         { \ctex_file_input:n { ctex-fontset-macold.def } }
5070         { \ctex_file_input:n { ctex-fontset-macnew.def } }
5071     }
5072 }
5073 </mac>

```

macold 的设置参考了 [OS X Mavericks \(10.9\) 预装的主要简体中文字体列表](#)。

macnew 在默认字体设置方面, 引入了多字重的宋体作为罗马字族, 以及引入了苹方黑体作为无衬线字族。由于 Songti SC Light 的字重与 STSong 及 Windows 上的 SimSun 更接近, 故默认字重使用 Songti SC Light, 而不带后缀的正常字重事实上没有使用。黑体、圆体等设置也有类似的情况。

```

5074 <*macold|macnew>
5075 \ctex_fontset_case:nnnn
5076 {
5077     \ctex_fontset_error:n { mac }
5078     \use_none_delimit_by_q_stop:w
5079 }
5080 <*macold>
5081 { \ctex_fontset_error:n { macold } }
5082 { \ctex_fontset_error:n { macold } }
5083 </macold>
5084 <*macnew>
5085 {
5086     \ctex_zhmap_case:nnn

```

```

5087 {
5088   \setCJKmainfont { :3:Songti.ttc }
5089   [
5090     BoldFont      = :1:Songti.ttc,
5091     ItalicFont    = :0:Kaiti.ttc,
5092     BoldItalicFont = :3:Kaiti.ttc,
5093   ]
5094   \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
5095   \setCJKmonofont { STFANGSO.ttf }
5096   \setCJKfamilyfont { zhsong } { :3:Songti.ttc } [ BoldFont = :1:Songti.ttc ]
5097   \setCJKfamilyfont { zhhei } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
5098   \setCJKfamilyfont { zhkai } { :0:Kaiti.ttc } [ BoldFont = :3:Kaiti.ttc ]
5099   \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
5100   \setCJKfamilyfont { zhli } { :0:Baoli.ttc }
5101   \setCJKfamilyfont { zhyou } { :4:Yuanti.ttc } [ BoldFont = :0:Yuanti.ttc ]
5102   \ctex_punct_set:n { mac }
5103   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5104   \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
5105   \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5106   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5107   \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
5108   \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhpfb }
5109   \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
5110 }
5111 {
5112   \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { mac }
5113   \ctex_punct_set:n { mac }
5114   \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5115   \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
5116   \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhpfb }
5117   \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5118 }
5119 { \ctex_fontset_error:n { macnew } }
5120 \use_none_delimit_by_q_stop:w
5121 }
5122 {
5123   \ctex_set_upmap:nnn { upserif } { :3:Songti.ttc } { :1:Songti.ttc }
5124   \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc } { }
5125   \ctex_set_upmap:nnn { upstsl } { :0:Baoli.ttc } { }
5126   \ctex_set_upmap:nnn { upstht } { :4:Yuanti.ttc } { }
5127   \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
5128   \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf } { }
5129   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5130   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5131   \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
5132   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
5133   \ctex_set_upfamily:nnn { zhli } { upschrn } { }
5134   \ctex_set_upfamily:nnn { zhyou } { upschtg } { }
5135   \use_none_delimit_by_q_stop:w
5136 }
5137 </macnew>
5138 {
5139 <*macold>
5140   \setCJKmainfont { STSong }
5141   [ BoldFont = STHeiti, ItalicFont = STKaiti ]
5142   \setCJKsansfont { STXihei } [ BoldFont = STHeiti ]
5143   \setCJKmonofont { STFangsong }
5144   \setCJKfamilyfont { zhsong } { STSong }
5145   \setCJKfamilyfont { zhhei } { STHeiti }
5146   \setCJKfamilyfont { zhfs } { STFangsong }
5147   \setCJKfamilyfont { zhkai } { STKaiti }
5148 </macold>
5149 <*macnew>
5150   \bool_new:N \g__ctex_mac_pingfang_bool

```

为方便人类阅读,把LuaTeX 分支挪到 `\ctex_fontset_case:nnnn` 之后,在不忽略空格的 cat-

code scheme 下编写。这样 Lua 代码就不必以~ 代替空格。macnew 的其他 \ctex\_fontset\_case:nnnn 和其他 \ctex\_zhmap\_case:nnn 分支，末尾均使用 \use\_none\_delimit\_by\_q\_stop:w 来跳过 LuaTeX 分支。

```

5151 \sys_if_engine luatex:F
5152 {
5153   \fontspec_font_if_exist:nTF { Kaiti~SC }
5154   {
5155     \setCJKmainfont { Songti~SC~Light }
5156     [
5157       BoldFont      = Songti~SC~Bold ,
5158       ItalicFont    = Kaiti~SC ,
5159       BoldItalicFont = Kaiti~SC~Bold ,
5160     ]
5161   }
5162   {
5163     \setCJKmainfont { Songti~SC~Light }
5164     [ BoldFont = Songti~SC~Bold ]
5165   }
5166   \setCJKfamilyfont { zhsong } { Songti~SC~Light }
5167   [ BoldFont = Songti~SC~Bold ]
5168   \setCJKfamilyfont { zhhei } { Heiti~SC~Light }
5169   [ BoldFont = Heiti~SC~Medium ]
5170   \fontspec_font_if_exist:nTF { PingFang~SC }
5171   {
5172     \setCJKsansfont { PingFang~SC }
5173     \setCJKfamilyfont { zhpj } { PingFang~SC }
5174     \bool_gset_true:N \g__ctex_mac_pingfang_bool
5175   }
5176   { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
5177   \fontspec_font_if_exist:nTF { STFangsong }
5178   {
5179     \setCJKmonofont { STFangsong }
5180     \setCJKfamilyfont { zhfs } { STFangsong }
5181   }
5182   { \setCJKmonofont { Heiti~SC~Light } }
5183   \fontspec_font_if_exist:nT { Kaiti~SC }
5184   {
5185     \setCJKfamilyfont { zhkai } { Kaiti~SC }
5186     [ BoldFont = Kaiti~SC~Bold ]
5187   }
5188   \fontspec_font_if_exist:nT { Baoli~SC }
5189   { \setCJKfamilyfont { zhli } { Baoli~SC } }
5190   \fontspec_font_if_exist:nT { Yuanti~SC~Light }
5191   {
5192     \setCJKfamilyfont { zhyou } { Yuanti~SC~Light }
5193     [ BoldFont = Yuanti~SC~Regular ]
5194   }
5195   \use_none_delimit_by_q_stop:w
5196 }
5197 </macnew>
5198 }
5199 <*macnew>
5200 \char_set_catcode_space:n { 32 }
5201 \lua_now:n
5202 {
5203   local lfs = require("lfs");
5204   local search_bases = {};
5205   local asset_roots = {
5206     "/System/Library/AssetsV2",
5207     "/System/Library/AssetsV2/PreinstalledAssetsV2/InstallWith0s",
5208   };
5209   for _, root in ipairs(asset_roots) do
5210     local ok, iter, obj = pcall(lfs.dir, root);
5211     if ok then
5212       for d in iter, obj do

```

```

5213         if d:find("^com_apple_MobileAsset_Font") then
5214             table.insert(search_bases, root .. "/" .. d);
5215         end;
5216     end;
5217 end;
5218 end;
5219 local function find_font_dir(filename)
5220     for _, base in ipairs(search_bases) do
5221         local ok, iter, obj = pcall(lfs.dir, base);
5222         if ok then
5223             for entry in iter, obj do
5224                 if not (entry == "." or entry == "..") then
5225                     local p = base .. "/" .. entry
5226                         .. "/AssetData/" .. filename;
5227                     if lfs.attributes(p, "mode") then
5228                         return base .. "/" .. entry .. "/AssetData/";
5229                     end;
5230                 end;
5231             end;
5232         end;
5233     end;
5234     return nil;
5235 end;
5236 local fonts = {
5237     {"PingFang.ttc", "g__ctex_mac_pingfang_dir_tl"},
5238     {"Kaiti.ttc", "g__ctex_mac_kaiti_dir_tl"},
5239     {"STFANGSO.ttf", "g__ctex_mac_stfangso_dir_tl"},
5240     {"Baoli.ttc", "g__ctex_mac_baoli_dir_tl"},
5241     {"Yuanti.ttc", "g__ctex_mac_yuanti_dir_tl"},
5242 };
5243 for _, f in ipairs(fonts) do
5244     local dir = find_font_dir(f[1]);
5245     token.set_macro(f[2], dir or "", "global");
5246 end
5247 }
5248 \char_set_catcode_ignore:n { 32 }
5249 \tl_if_empty:NTF \g__ctex_mac_kaiti_dir_tl
5250 {
5251     \setCJKmainfont { Songti~SC~Light }
5252     [ BoldFont = Songti~SC~Bold ]
5253 }
5254 {
5255     \setCJKmainfont { Songti~SC~Light }
5256     [
5257         BoldFont      = Songti~SC~Bold ,
5258         ItalicFont    = Kaiti.ttc ,
5259         ItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 0 } ,
5260         BoldItalicFont = Kaiti.ttc ,
5261         BoldItalicFeatures = { Path = \g__ctex_mac_kaiti_dir_tl , FontIndex = 3 } ,
5262     ]
5263 }
5264 \setCJKfamilyfont { zhsong } { Songti~SC~Light }
5265 [ BoldFont = Songti~SC~Bold ]
5266 \setCJKfamilyfont { zhhei } { Heiti~SC~Light }
5267 [ BoldFont = Heiti~SC~Medium ]
5268 \tl_if_empty:NTF \g__ctex_mac_pingfang_dir_tl
5269 { \setCJKsansfont { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ] }
5270 {
5271     \setCJKsansfont { PingFang.ttc }
5272     [
5273         Path          = \g__ctex_mac_pingfang_dir_tl ,
5274         FontIndex     = 2 ,
5275         BoldFont      = PingFang.ttc ,
5276         BoldFontIndex = 8 ,
5277     ]
5278 \setCJKfamilyfont { zhpf } { PingFang.ttc }
5279 [ Path = \g__ctex_mac_pingfang_dir_tl , FontIndex = 2 ]

```

```

5280 \bool_gset_true:N \g__ctex_mac_pingfang_bool
5281 }
5282 \tl_if_empty:NTF \g__ctex_mac_stfangso_dir_tl
5283 { \setCJKmonofont { Heiti~SC~Light } }
5284 {
5285 \setCJKmonofont { STFANGSO.ttf }
5286 [ Path = \g__ctex_mac_stfangso_dir_tl ]
5287 \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
5288 [ Path = \g__ctex_mac_stfangso_dir_tl ]
5289 }
5290 \tl_if_empty:NF \g__ctex_mac_kaiti_dir_tl
5291 {
5292 \setCJKfamilyfont { zhkai } { Kaiti.ttc }
5293 [
5294 Path = \g__ctex_mac_kaiti_dir_tl ,
5295 FontIndex = 0 ,
5296 BoldFont = Kaiti.ttc ,
5297 BoldFontIndex = 3 ,
5298 ]
5299 }
5300 \tl_if_empty:NF \g__ctex_mac_baoli_dir_tl
5301 {
5302 \setCJKfamilyfont { zhli } { Baoli.ttc }
5303 [ Path = \g__ctex_mac_baoli_dir_tl , FontIndex = 0 ]
5304 }
5305 \tl_if_empty:NF \g__ctex_mac_yuanti_dir_tl
5306 {
5307 \setCJKfamilyfont { zhyou } { Yuanti.ttc }
5308 [
5309 Path = \g__ctex_mac_yuanti_dir_tl ,
5310 FontIndex = 4 ,
5311 BoldFont = Yuanti.ttc ,
5312 BoldFontIndex = 0 ,
5313 ]
5314 }
5315 \use_none_delimit_by_q_stop:w \q_stop
5316 </macnew>
5317 </macold|macnew>

```

### 15.15.2.6 ubuntu

```

5318 <*ubuntu>
5319 \ctex_fontset_case:nnnn
5320 { \ctex_fontset_error:n { ubuntu } }
5321 {
5322 \ctex_zhmap_case:nnn
5323 {
5324 \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
5325 [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
5326 \setCJKsansfont { :2:NotoSansCJK-Regular.ttc }
5327 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
5328 \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
5329 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
5330 \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
5331 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
5332 \setCJKfamilyfont { zhhei } { :2:NotoSansCJK-Regular.ttc }
5333 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
5334 \setCJKfamilyfont { zhkai } { gkai00mp.ttf }
5335 \ctex_punct_set:n { ubuntu }
5336 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5337 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
5338 \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
5339 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5340 \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
5341 \ctex_punct_map_bfseries:nn
5342 { \CJKrmdefault, \CJKttdefault, zhsong }

```

```

5343     { zhsongb }
5344   }
5345   {
5346     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { ubuntu }
5347     \ctex_punct_set:n { ubuntu }
5348     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5349     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5350     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5351   }
5352   { \ctex_fontset_error:n { ubuntu } }
5353 }
5354 {
5355   \ctex_set_upmap_unicode:nnn { upserif }
5356   { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
5357   \ctex_set_upmap_unicode:nnn { upsans }
5358   { :2:NotoSansCJK-Regular.ttc } { :2:NotoSansCJK-Bold.ttc }
5359   \ctex_set_upmap_unicode:nnn { upmono }
5360   { :2:NotoSerifCJK-Regular.ttc } { }
5361   \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf } { }
5362   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
5363   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
5364   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
5365 }
5366 {
5367   \setCJKmainfont { Noto~Serif~CJK~SC }
5368   [ BoldFont = Noto~Serif~CJK~SC~Bold, ItalicFont = AR~PL~KaitiM~GB ]
5369   \setCJKsansfont { Noto~Sans~CJK~SC }
5370   [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5371   \setCJKmonofont { Noto~Serif~CJK~SC }
5372   \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
5373   [ BoldFont = Noto~Serif~CJK~SC~Bold ]
5374   \setCJKfamilyfont { zhhei } { Noto~Sans~CJK~SC }
5375   [ BoldFont = Noto~Sans~CJK~SC~Bold ]
5376   \setCJKfamilyfont { zhkai } { AR~PL~KaitiM~GB }
5377 }
5378 </ubuntu>

```

### 15.15.2.7 windows

\c\_\_ctex\_msyh\_suffix\_tl Windows 8 以后, 微软雅黑由原来的 .ttf 后缀改为 .ttc 后缀, 需要加以区分。

```

5379 <*windows>
5380 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
5381 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5382 {
5383   \file_if_exist:nTF { msyh.ttc }
5384   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
5385   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
5386 }
5387 \ctex_fontset_case:nnn
5388 {
5389   \ctex_zhmap_case:nnn
5390   {
5391     \ctex_punct_set:n { windows }
5392     \setCJKmainfont { simsun.ttc }
5393     [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
5394     \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
5395     [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5396     \setCJKmonofont { simfang.ttf }
5397     \setCJKfamilyfont { zhsong } { simsun.ttc }
5398     \setCJKfamilyfont { zhhei } { simhei.ttf }
5399     \setCJKfamilyfont { zhfs } { simfang.ttf }
5400     \setCJKfamilyfont { zhkai } { simkai.ttf }
5401     \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
5402     [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
5403     \setCJKfamilyfont { zhli } { simli.ttf }

```

```

5404     \setCJKfamilyfont { zhyou } { simyou.ttf }
5405     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5406     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5407     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5408     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
5409     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
5410     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
5411 }
5412 {
5413     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { windows }
5414     \ctex_punct_set:n { windows }
5415     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
5416     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
5417     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
5418 }
5419 { \ctex_fontset_error:n { windows } }
5420 }
5421 {
5422     \ctex_set_upfonts:nnnnnn
5423     { simsun.ttc }
5424     { simhei.ttf }
5425     { simkai.ttf }
5426     { msyh.\c__ctex_msyh_suffix_tl }
5427     { msyhbd.\c__ctex_msyh_suffix_tl }
5428     { simfang.ttf }
5429     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
5430     \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
5431     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
5432     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
5433     \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
5434     \ctex_set_upfamily:nnn { zhli } { upschrn } {}
5435     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
5436     \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
5437     \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
5438 }
5439 {
5440     \setCJKmainfont { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
5441     \setCJKsansfont { MicrosoftYaHei } [ BoldFont = *~Bold ]
5442     \setCJKmonofont { FangSong }
5443     \setCJKfamilyfont { zhsong } { SimSun }
5444     \setCJKfamilyfont { zhhei } { SimHei }
5445     \setCJKfamilyfont { zhfs } { FangSong }
5446     \setCJKfamilyfont { zhkai } { KaiTi }
5447     \setCJKfamilyfont { zhyahei } { MicrosoftYaHei } [ BoldFont = *~Bold ]
5448     \setCJKfamilyfont { zhli } { LiSu }
5449     \setCJKfamilyfont { zhyou } { YouYuan }
5450 }
5451 </windows>

```

### 15.15.3 中文字体命令

使用 $\text{upLaTeX}$ 编译时, macnew 字库中由于传统黑体(黑体-简)无法使用, 我们用苹方来代替。同时 `\yahei`、`\pingfang` 命令被设置为与 `\heiti` 相同。

```

\songti
\heiti
\fangsong
\kaishu
\lishu
\youyuan
\yahei
\pingfang
5452 < *!mac >
5453 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
5454 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
5455 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
5456 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
5457 < *windows|founder|macnew >
5458 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
5459 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
5460 < *windows|founder|macnew >
< windows > 5461 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
5462 < *macnew >

```

```

5463 \bool_lazy_or:nnTF
5464 { \sys_if_engine_pdftex_p: }
5465 { \sys_if_engine_uptex_p: }
5466 {
5467   \cs_new_eq:NN \yahei \heiti
5468   \cs_new_eq:NN \pingfang \heiti
5469 }
5470 {
5471   \bool_if:NTF \g__ctex_mac_pingfang_bool
5472   {
5473     \NewDocumentCommand \yahei { } { \CJKfamily { zhp } }
5474     \NewDocumentCommand \pingfang { } { \CJKfamily { zhp } }
5475   }
5476   {
5477     \cs_new_eq:NN \yahei \heiti
5478     \cs_new_eq:NN \pingfang \heiti
5479   }
5480 }
5481 </macnew>
5482 </!mac>

5483 </fontset>

```

#### 15.15.4 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

5484 <zhmap>
5485 \begingroup\catcode61\catcode48\catcode32=10\relax%
5486 \catcode 35=6 % #
5487 \catcode 45=12 % -
5488 \catcode123=1 % {
5489 \catcode125=2 % }
5490 \toks0{\endlinechar=\the\endlinechar\relax}%
5491 \toks2{\endlinechar=-1}%
5492 \def\x#1 #2 {%
5493   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
5494   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
5495 \x 13 5 % carriage return
5496 \x 32 10 % space
5497 \x 35 6 % #
5498 \x 40 12 % (
5499 \x 41 12 % )
5500 \x 45 12 % -
5501 \x 46 12 % .
5502 \x 47 12 % /
5503 \x 58 12 % :
5504 \x 60 12 % <
5505 \x 61 12 % =
5506 \x 64 11 % @
5507 \x 91 12 % [
5508 \x 93 12 % ]
5509 \x 123 1 % {
5510 \x 125 2 % }
5511 \edef\x#1{\endgroup%
5512   \edef\noexpand#1{%
5513     \the\toks0 %
5514     \let\noexpand\noexpand\noexpand#1%
5515     \noexpand\noexpand\noexpand\noexpand\undefined%
5516     \noexpand\noexpand\noexpand\noexpand\endinput}%
5517   \the\toks2}%
5518 \expandafter\x\csname ctex@zhmap@endinput\endcsname

\ifzhmappdf 5519 \begingroup\expandafter\endgroup
5520 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
5521 \expandafter\ifx\csname ifctexpdf\endcsname\relax

```

```

5522 \expandafter\ifx\csname pdfoutput\endcsname\relax
5523     iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
5524 \else ifctexpdf\fi
5525 \endcsname

```

\ProvidesFile 提供非 L<sup>A</sup>T<sub>E</sub>X 格式下的 \ProvidesFile。

```

5526 \begingroup
5527 \expandafter\ifx\csname ProvidesFile\endcsname\relax
5528 \long\def\x#1\ProvidesFile#2[#3]{%
5529     #1%
5530     \immediate\write-1{File: #2 #3}%
5531     \expandafter\xdef\csname ver@#2\endcsname{#3}}
5532 \expandafter\x%
5533 \fi
5534 \endgroup

```

文件标识信息。

```

<adobe> 5535 \ProvidesFile{ctex-zhmap-adobe.tex}%
<adobe> 5536 [2022/07/14 v2.5.10 Adobe font map loader for DVIPDFMx (CTEX)]
<fandol> 5537 \ProvidesFile{ctex-zhmap-fandol.tex}%
<fandol> 5538 [2022/07/14 v2.5.10 Fandol font map loader for DVIPDFMx (CTEX)]
<founder> 5539 \ProvidesFile{ctex-zhmap-founder.tex}%
<founder> 5540 [2022/07/14 v2.5.10 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
<hanyi> 5541 \ProvidesFile{ctex-zhmap-hanyi.tex}%
<hanyi> 5542 [2022/07/14 v2.5.10 Hanyi font map loader for pdfTeX and DVIPDFMx (CTEX)]
<mac> 5543 \ProvidesFile{ctex-zhmap-mac.tex}%
<mac> 5544 [2022/07/14 v2.5.10 Mac font map loader for DVIPDFMx (CTEX)]
<ubuntu> 5545 \ProvidesFile{ctex-zhmap-ubuntu.tex}%
<ubuntu> 5546 [2022/07/14 v2.5.10 Ubuntu font map loader for DVIPDFMx (CTEX)]
<windows> 5547 \ProvidesFile{ctex-zhmap-windows.tex}%
<windows> 5548 [2022/07/14 v2.5.10 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]

```

#### 15.15.4.1 ctex-zhmap-adobe.tex

```

5549 <*adobe>
5550 \ifzhmappdf
5551 %% pdfTeX does not support OTF fonts
5552 \else
5553 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5554 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5555 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
5556 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
5557 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
5558 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5559 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
5560 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
5561 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
5562 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
5563 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
5564 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
5565 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
5566 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
5567 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
5568 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
5569 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
5570 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
5571 \fi
5572 </adobe>

```

#### 15.15.4.2 ctex-zhmap-fandol.tex

```

5573 <*fandol>
5574 \ifzhmappdf
5575 %% pdfTeX does not support OTF fonts

```

```

5576 \else
5577   \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
5578   \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
5579   \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
5580   \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
5581   \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
5582   \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
5583   \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
5584   \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
5585   \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
5586   \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
5587   \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
5588   \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
5589   \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
5590   \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
5591   \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
5592   \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
5593   \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
5594   \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
5595 \fi
5596 </fandol>

```

### 15.15.4.3 ctex-zhmap-founder.tex

```

5597 <*founder>
5598 \ifzhmappdf
5599   \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
5600   \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
5601   \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
5602   \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
5603   \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
5604   \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
5605   \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
5606   \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
5607   \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
5608   \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
5609   \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
5610   \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
5611   \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
5612   \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
5613   \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
5614   \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
5615   \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
5616   \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
5617   \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
5618   \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
5619   \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
5620   \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
5621   \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
5622   \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
5623   \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
5624   \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
5625 \else
5626   \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
5627   \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
5628   \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
5629   \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
5630   \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
5631   \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
5632   \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
5633   \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
5634   \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
5635   \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
5636   \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
5637   \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
5638   \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
5639   \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF}

```

```

5640 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
5641 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
5642 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
5643 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
5644 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
5645 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
5646 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
5647 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
5648 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
5649 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
5650 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
5651 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
5652 \fi
5653 </founder>

```

#### 15.15.4.4 ctex-zhmap-hanyi.tex

```

5654 <*hanyi>
5655 \ifzhmappdf
5656 \pdfmapline{=gbk@UGBK@ <HYShuSongErS.ttf}
5657 \pdfmapline{=gbksong@UGBK@ <HYShuSongErS.ttf}
5658 \pdfmapline{=gbkkai@UGBK@ <HYKaiTiS.ttf}
5659 \pdfmapline{=gbkhei@UGBK@ <HYZhongHeiS.ttf}
5660 \pdfmapline{=gbkfs@UGBK@ <HYFangSongS.ttf}
5661 \pdfmapline{=cyberb@Unicode@ <HYShuSongErS.ttf}
5662 \pdfmapline{=unisong@Unicode@ <HYShuSongErS.ttf}
5663 \pdfmapline{=unikai@Unicode@ <HYKaiTiS.ttf}
5664 \pdfmapline{=unihei@Unicode@ <HYZhongHeiS.ttf}
5665 \pdfmapline{=unifs@Unicode@ <HYFangSongS.ttf}
5666 \pdfmapline{=gbksongsl@UGBK@ <HYShuSongErS.ttf}
5667 \pdfmapline{=gbkkaisl@UGBK@ <HYKaiTiS.ttf}
5668 \pdfmapline{=gbkheisl@UGBK@ <HYZhongHeiS.ttf}
5669 \pdfmapline{=gbkfssl@UGBK@ <HYFangSongS.ttf}
5670 \pdfmapline{=unisongsl@Unicode@ <HYShuSongErS.ttf}
5671 \pdfmapline{=unikaisl@Unicode@ <HYKaiTiS.ttf}
5672 \pdfmapline{=uniheisl@Unicode@ <HYZhongHeiS.ttf}
5673 \pdfmapline{=unifssl@Unicode@ <HYFangSongS.ttf}
5674 \else
5675 \special{pdf:mapline gbk@UGBK@ unicode HYShuSongErS.ttf}
5676 \special{pdf:mapline gbksong@UGBK@ unicode HYShuSongErS.ttf}
5677 \special{pdf:mapline gbkkai@UGBK@ unicode HYKaiTiS.ttf}
5678 \special{pdf:mapline gbkhei@UGBK@ unicode HYZhongHeiS.ttf}
5679 \special{pdf:mapline gbkfs@UGBK@ unicode HYFangSongS.ttf}
5680 \special{pdf:mapline cyberb@Unicode@ unicode HYShuSongErS.ttf}
5681 \special{pdf:mapline unisong@Unicode@ unicode HYShuSongErS.ttf}
5682 \special{pdf:mapline unikai@Unicode@ unicode HYKaiTiS.ttf}
5683 \special{pdf:mapline unihei@Unicode@ unicode HYZhongHeiS.ttf}
5684 \special{pdf:mapline unifs@Unicode@ unicode HYFangSongS.ttf}
5685 \special{pdf:mapline gbksongsl@UGBK@ unicode HYShuSongErS.ttf -s .167}
5686 \special{pdf:mapline gbkkaisl@UGBK@ unicode HYKaiTiS.ttf -s .167}
5687 \special{pdf:mapline gbkheisl@UGBK@ unicode HYZhongHeiS.ttf -s .167}
5688 \special{pdf:mapline gbkfssl@UGBK@ unicode HYFangSongS.ttf -s .167}
5689 \special{pdf:mapline unisongsl@Unicode@ unicode HYShuSongErS.ttf -s .167}
5690 \special{pdf:mapline unikaisl@Unicode@ unicode HYKaiTiS.ttf -s .167}
5691 \special{pdf:mapline uniheisl@Unicode@ unicode HYZhongHeiS.ttf -s .167}
5692 \special{pdf:mapline unifssl@Unicode@ unicode HYFangSongS.ttf -s .167}
5693 \fi
5694 </hanyi>

```

#### 15.15.4.5 ctex-zhmap-mac.tex

```

5695 <*mac>
5696 \ifzhmappdf
5697 %% pdfTeX does not support OTF fonts
5698 \else
5699 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
5700 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H :3:Songti.ttc}

```

```

5701 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc}
5702 \special{pdf:mapline gbkhei@UGBK@ unicode :2:PingFang.ttc}
5703 \special{pdf:mapline gbkfs@UGBK@ unicode STFANGSO.ttf}
5704 \special{pdf:mapline gbkli@UGBK@ UniGB-UTF16-H :0:Baoli.ttc}
5705 \special{pdf:mapline gbkyou@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc}
5706 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
5707 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
5708 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc}
5709 \special{pdf:mapline unihei@Unicode@ unicode :2:PingFang.ttc}
5710 \special{pdf:mapline unifs@Unicode@ unicode STFANGSO.ttf}
5711 \special{pdf:mapline unili@Unicode@ UniGB-UTF16-H :0:Baoli.ttc}
5712 \special{pdf:mapline uniyou@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc}
5713 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H :3:Songti.ttc -s .167}
5714 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
5715 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:PingFang.ttc -s .167}
5716 \special{pdf:mapline gbkfssl@UGBK@ unicode STFANGSO.ttf -s .167}
5717 \special{pdf:mapline gbklisl@UGBK@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
5718 \special{pdf:mapline gbkyousl@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
5719 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc -s .167}
5720 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
5721 \special{pdf:mapline uniheisl@Unicode@ unicode :2:PingFang.ttc -s .167}
5722 \special{pdf:mapline unifssl@Unicode@ unicode STFANGSO.ttf -s .167}
5723 \special{pdf:mapline unilisl@Unicode@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
5724 \special{pdf:mapline uniyousl@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
5725 \fi
5726 </mac>

```

#### 15.15.4.6 ctex-zhmap-ubuntu.tex

```

5727 <*ubuntu>
5728 \ifzhmappdf
5729 %% pdfTeX does not support OTF fonts
5730 \else
5731 \special{pdf:mapline gbk@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5732 \special{pdf:mapline gbksong@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5733 \special{pdf:mapline gbkkai@UGBK@ unicode gkai00mp.ttf}
5734 \special{pdf:mapline gbkhei@UGBK@ unicode :2:NotoSansCJK-Regular.ttc}
5735 \special{pdf:mapline gbkfs@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5736 \special{pdf:mapline cyberb@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5737 \special{pdf:mapline unisong@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5738 \special{pdf:mapline unikai@Unicode@ unicode gkai00mp.ttf}
5739 \special{pdf:mapline unihei@Unicode@ unicode :2:NotoSansCJK-Regular.ttc}
5740 \special{pdf:mapline unifs@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5741 \special{pdf:mapline gbksongsl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5742 \special{pdf:mapline gbkkaisl@UGBK@ unicode gkai00mp.ttf -s .167}
5743 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
5744 \special{pdf:mapline gbkfssl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5745 \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5746 \special{pdf:mapline unikaisl@Unicode@ unicode gkai00mp.ttf -s .167}
5747 \special{pdf:mapline uniheisl@Unicode@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
5748 \special{pdf:mapline unifssl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5749 \fi
5750 </ubuntu>

```

#### 15.15.4.7 ctex-zhmap-windows.tex

```

5751 <*windows>
5752 \ifzhmappdf
5753 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
5754 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
5755 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
5756 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
5757 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
5758 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
5759 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
5760 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
5761 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
5762 \pdfmapline{=unikai@Unicode@ <simkai.ttf}

```

```

5763 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
5764 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
5765 \pdfmapline{=unili@Unicode@ <simli.ttf}
5766 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
5767 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
5768 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
5769 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
5770 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
5771 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
5772 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
5773 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
5774 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
5775 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
5776 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
5777 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
5778 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
5779 \else
5780 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
5781 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
5782 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
5783 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
5784 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
5785 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
5786 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
5787 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
5788 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
5789 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
5790 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
5791 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
5792 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
5793 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
5794 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -v 50 -s .167}
5795 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -v 70 -s .167}
5796 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -v 150 -s .167}
5797 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -v 50 -s .167}
5798 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -v 150 -s .167}
5799 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -v 60 -s .167}
5800 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -v 50 -s .167}
5801 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -v 70 -s .167}
5802 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -v 150 -s .167}
5803 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -v 50 -s .167}
5804 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -v 150 -s .167}
5805 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -v 60 -s .167}
5806 \fi
5807 </windows>

5808 \ctex@zhmap@endinput
5809 </zhmap>

```

### 15.15.5 制作 spa 文件

我们通过 X<sub>Y</sub>TeX 的 \XeTeXglyphbounds 取得字体中标点符号的边界信息, 为 CJKpunct 宏包制作 spa。

```

5810 <*spa>
5811 <*macro>
5812 \input expl3-generic %
5813 \ExplSyntaxOn
5814 \sys_if_engine_xetex:F
5815 {
5816   \msg_new:nnn { ctex } { xetex }
5817   { XeTeX~is~required~to~compile~this~document! }
5818   \msg_fatal:nn { ctex } { xetex }
5819 }

```

CJKpunct 定义的标点符号是:

```
‘ “ 「 『 { ( [ { < 《 [ [
—…、。、. : ; ! ? % ) ] ] } > » ] ] ’ ” 」 』
```

注意顺序不能改变。

```
5820 \seq_const_from_clist:Nn \c__ctex_punct_seq
5821 {
5822   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
5823   "3008 , "300A , "3016 , "3010 ,
5824   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
5825   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
5826   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
5827 }
```

`\ctex_make_spa:nn` #1 是 spa 文件名, #2 是由 CJK 族名与字体构成的逗号列表。

```
5828 \cs_new_protected:Npn \ctex_make_spa:nn #1#2
5829 {
5830   \iow_open:Nn \g__ctex_spa_iow {#1}
5831   \clist_map_inline:nn {#2}
5832     { \__ctex_write_family:nn ##1 }
5833   \iow_close:N \g__ctex_spa_iow
5834 }
5835 \iow_new:N \g__ctex_spa_iow
5836 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

5837 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
5838 {
5839   \group_begin:
5840   \__ctex_if_font_exists:nTF {#2}
5841     { \tex_font:D \l__ctex_punct_font = "[#2]" ~ at ~ 100 pt \scan_stop: }
5842     { \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop: }
5843   \l__ctex_punct_font
5844   \clist_clear:N \l__ctex_punct_bounds_clist
5845   \seq_map_inline:Nn \c__ctex_punct_seq
5846     {
5847       \exp_args:No \__ctex_save_bounds:n
5848       { \int_use:N \tex_XeTeXcharglyph:D ##1 }
5849     }
5850   \iow_now:Ne \g__ctex_spa_iow
5851   {
5852     \token_to_str:N \ctexspadef {#1}
```

最后这三个逗号对 `CJKpunct` 来说是必要的。

```
5853     { \l__ctex_punct_bounds_clist , , , }
5854   }
5855   \group_end:
5856 }
```

`\__ctex_if_font_exists:nTF` 优先搜索字体是否为  $\TeX$  Tree 字体或是否存在于当前路径下<sup>21</sup>, 如果不存在, `\__ctex_write_family:nn` 会根据结果在 shell 中调用 `kpathsea` 查文件或 `fontconfig` 查名称<sup>22</sup>。

```
5857 \prg_new_protected_conditional:Nnn \__ctex_if_font_exists:n { TF }
5858 {
5859   \tl_clear:N \l__ctex_if_otf_exists_tl
5860   \tl_clear:N \l__ctex_if_ttf_exists_tl
5861   \bool_case:nF
5862     {
5863       { \file_if_exist_p:n { #1.otf } }
5864       { \tl_set:Nn \l__ctex_if_otf_exists_tl {#1} }
5865       { \file_if_exist_p:n { #1.ttf } }
```

<sup>21</sup>事实上 `TEXINPUTS` 也会被一并优先搜索。

<sup>22</sup>如果用户把字体目录加入到了 `TEXINPUTS` 中, `\file_if_exist_p:n` 可能找到一个 `XqTeX [basename]` 加载不了的文件, 因为 `XqTeX` 搜索字体专属路径 `OPENTYPEFONTS/ TTFONTS`, 不搜索 `TEXINPUTS`。

```

5866     { \tl_set:Nn \l__ctex_if_ttf_exists_tl {#1} }
5867   }
5868   {
5869     \sys_get_shell:nnN { kpsewhich ~ #1.otf } { \endlinechar=-1 }
5870     \l__ctex_if_otf_exists_tl
5871     \sys_get_shell:nnN { kpsewhich ~ #1.ttf } { \endlinechar=-1 }
5872     \l__ctex_if_ttf_exists_tl
5873   }
5874   \bool_lazy_and:nnTF
5875     { \tl_if_blank_p:V \l__ctex_if_otf_exists_tl }
5876     { \tl_if_blank_p:V \l__ctex_if_ttf_exists_tl }
5877     { \prg_return_false: } { \prg_return_true: }
5878 }

```

\l\_\_ctex\_if\_otf\_exists\_tl 用于存储shell 输出的kpsewhich 结果。

\l\_\_ctex\_if\_ttf\_exists\_tl

```

5879 \tl_clear_new:N \l__ctex_if_otf_exists_tl
5880 \tl_clear_new:N \l__ctex_if_ttf_exists_tl

5881 \cs_new_protected:Npn \__ctex_save_bounds:n #1
5882 {
5883   \clist_put_right:Ne \l__ctex_punct_bounds_clist
5884   {
5885     \__ctex_calc_bounds:nn { 1 } {#1} ,
5886     \__ctex_calc_bounds:nn { 3 } {#1}
5887   }
5888 }
5889 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1 em 的比值的一百倍。

```

5890 \cs_new:Npn \__ctex_calc_bounds:nn #1#2
5891 {
5892   \fp_eval:n
5893   {
5894     round
5895     (
5896       \dim_to_decimal_in_unit:nn
5897       { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
5898       { 1 em }
5899     )
5900   }
5901 }
5902 \ExplSyntaxOff
5903 </macro>

```

下面是 C<sub>T</sub>E<sub>X</sub> 定义的一些字体。

```

5904 <*make>
5905 \input ctex-spa-macro %
5906 \MAKESPA {ctexpunct.spa}
5907 {
5908   {adobezhsong}      {AdobeSongStd-Light} ,
5909   {adobezhhei}      {AdobeHeitiStd-Regular} ,
5910   {adobezhkai}      {AdobeKaitiStd-Regular} ,
5911   {adobezhfs}       {AdobeFangsongStd-Regular} ,
5912 %
5913   {fandolzhsong}    {FandolSong-Regular} ,
5914   {fandolzhsongb}   {FandolSong-Bold} ,
5915   {fandolzhhei}     {FandolHei-Regular} ,
5916   {fandolzhheib}    {FandolHei-Bold} ,
5917   {fandolzhkai}     {FandolKai-Regular} ,
5918   {fandolzhfs}      {FandolFang-Regular} ,
5919 %
5920   {founderzhsong}   {FZShuSong-Z01} ,
5921   {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
5922   {founderzhhei}    {FZHei-B01} ,

```

```

5923 {founderzhheil} {FZXiHeiI-Z08} ,
5924 {founderzhkai} {FZKai-Z03} ,
5925 {founderzhfs} {FZFangSong-Z02} ,
5926 {founderzhli} {FZLiShu-S01} ,
5927 {founderzhyou} {FZXiYuan-M01} ,
5928 {founderzhyoub} {FZZhunYuan-M02} ,
5929 %
5930 {hanyizhsong} {HYShuSongEr S} ,
5931 {hanyizhsongb} {HYZhongSong S} ,
5932 {hanyizhhei} {HYZhongHei S} ,
5933 {hanyizhheib} {HYDaHei S} ,
5934 {hanyizhkai} {HYKaiTi S} ,
5935 {hanyizhfs} {HYFangSong S} ,
5936 %
5937 {maczhsong} {Songti SC Light} ,
5938 {maczhsongb} {Songti SC Bold} ,
5939 {maczhhei} {Heiti SC Medium} ,
5940 {maczhheil} {Heiti SC Light} ,
5941 {maczhkai} {Kaiti SC} ,
5942 {maczhkaib} {Kaiti SC Bold} ,
5943 {maczhfs} {STFangsong} ,
5944 {maczhli} {Baoli SC} ,
5945 {maczhyou} {Yuanti SC Light} ,
5946 {maczhyoub} {Yuanti SC Regular} ,
5947 {maczhpfb} {PingFang SC} ,
5948 {maczhpfb} {PingFang SC Semibold} ,
5949 %
5950 {ubuntuzhsong} {Noto Serif CJK SC} ,
5951 {ubuntuzhsongb} {Noto Serif CJK SC Bold} ,
5952 {ubuntuzhhei} {Noto Sans CJK SC} ,
5953 {ubuntuzhheib} {Noto Sans CJK SC Bold} ,
5954 {ubuntuzhkai} {AR PL KaitiM GB} ,
5955 %
5956 {windowszhsong} {SimSun} ,
5957 {windowszhhei} {SimHei} ,
5958 {windowszhkai} {KaiTi} ,
5959 {windowszhfs} {FangSong} ,
5960 {windowszhli} {LiSu} ,
5961 {windowszhyou} {YouYuan} ,
5962 {windowszhyahei} {Microsoft YaHei} ,
5963 {windowszhyaheib} {Microsoft YaHei Bold}
5964 }
5965 \primitive\end
5966 </make>
5967 </spa>

```

## 15.16 translator 宏包的中文字典

```
5968 <*dict>
```

包括 ChineseGBK 和 ChineseUTF8 两种形式, 目前只翻译 beamer 宏包需要的定理环境名称。

```

5969 <*theorem>
5970 \providetranslation{Comments}{评论}
5971 \providetranslation{comments}{评论}
5972 \providetranslation{Comment}{评论}
5973 \providetranslation{comment}{评论}
5974 \providetranslation{Corollaries}{推论}
5975 \providetranslation{corollaries}{推论}
5976 \providetranslation{Corollary}{推论}
5977 \providetranslation{corollary}{推论}
5978 \providetranslation{Definitions}{定义}
5979 \providetranslation{definitions}{定义}
5980 \providetranslation{Definition}{定义}
5981 \providetranslation{definition}{定义}

```

```

5982 \providetranslation{Examples}{例}
5983 \providetranslation{examples}{例}
5984 \providetranslation{Example}{例}
5985 \providetranslation{example}{例}
5986 \providetranslation{Exercises}{练习}
5987 \providetranslation{exercises}{练习}
5988 \providetranslation{Exercise}{练习}
5989 \providetranslation{exercise}{练习}
5990 \providetranslation{Facts}{事实}
5991 \providetranslation{facts}{事实}
5992 \providetranslation{Fact}{事实}
5993 \providetranslation{fact}{事实}
5994 \providetranslation{Key Lemmas}{关键引理}
5995 \providetranslation{key lemmas}{关键引理}
5996 \providetranslation{Key Lemma}{关键引理}
5997 \providetranslation{key lemma}{关键引理}
5998 \providetranslation{Key Observations}{关键观察}
5999 \providetranslation{key observations}{关键观察}
6000 \providetranslation{Key Observation}{关键观察}
6001 \providetranslation{key observation}{关键观察}
6002 \providetranslation{Lemmas}{引理}
6003 \providetranslation{lemmas}{引理}
6004 \providetranslation{Lemma}{引理}
6005 \providetranslation{lemma}{引理}
6006 \providetranslation{Main Theorems}{主要定理}
6007 \providetranslation{main theorems}{主要定理}
6008 \providetranslation{Main Theorem}{主要定理}
6009 \providetranslation{main theorem}{主要定理}
6010 \providetranslation{Observations}{观察}
6011 \providetranslation{observations}{观察}
6012 \providetranslation{Observation}{观察}
6013 \providetranslation{observation}{观察}
6014 \providetranslation{Problems}{问题}
6015 \providetranslation{problems}{问题}
6016 \providetranslation{Problem}{问题}
6017 \providetranslation{problem}{问题}
6018 \providetranslation{Proofs}{证明}
6019 \providetranslation{proofs}{证明}
6020 \providetranslation{Proof}{证明}
6021 \providetranslation{proof}{证明}
6022 \providetranslation{Proof Sketch}{证明提要}
6023 \providetranslation{Proof sketch}{证明提要}
6024 \providetranslation{proof sketch}{证明提要}
6025 \providetranslation{Proof Sketches}{证明提要}
6026 \providetranslation{Proof sketches}{证明提要}
6027 \providetranslation{proof sketches}{证明提要}
6028 \providetranslation{Sketch of Proof}{证明提要}
6029 \providetranslation{Sketch of Proofs}{证明提要}
6030 \providetranslation{Sketch of proof}{证明提要}
6031 \providetranslation{Sketch of proofs}{证明提要}
6032 \providetranslation{sketch of proof}{证明提要}
6033 \providetranslation{sketch of proofs}{证明提要}
6034 \providetranslation{Propositions}{命题}
6035 \providetranslation{propositions}{命题}
6036 \providetranslation{Proposition}{命题}
6037 \providetranslation{proposition}{命题}
6038 \providetranslation{Remarks}{注}
6039 \providetranslation{remarks}{注}
6040 \providetranslation{Remark}{注}
6041 \providetranslation{remark}{注}
6042 \providetranslation{Solutions}{解}
6043 \providetranslation{solutions}{解}
6044 \providetranslation{Solution}{解}
6045 \providetranslation{solution}{解}
6046 \providetranslation{Theorems}{定理}
6047 \providetranslation{theorems}{定理}
6048 \providetranslation{Theorem}{定理}

```

```
6049 \providetranslation{theorem}{定理}
6050 </theorem>
6051 </dict>
```

## 15.17 ctexcap 宏包

```
6052 <*ctexcap>
```

ctexcap 是过时宏包。

```
6053 \clist_new:N \l__ctex_ctexcap_options_clist
6054 \clist_set:Ne \l__ctex_ctexcap_options_clist
6055 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
6056 \msg_new:nnn { ctexcap } { deprecated }
6057 {
6058   Package~`ctexcap'~is~deprecated.\\
6059   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
6060   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
6061 }
6062 \msg_warning:nne { ctexcap } { deprecated }
6063 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }
```

ctexcap 是默认打开 heading 选项的 ctex。

```
6064 \PassOptionsToPackage { heading = true } { ctexcap }
6065 \RequirePackageWithOptions { ctex }
6066 </ctexcap>
```

## 15.18 ctexhook 宏包

```
6067 <*ctexhook>
```

\ctex\_if\_format\_at\_least:nTF 与 \IfFormatAtLeastTF 同义。

```
6068 \cs_new:Npn \ctex_if_format_at_least:nTF
6069 { \@ifl@t@r \fmtversion }
```

\ctex\_file\_input:n 输入文件, 关闭 L<sup>A</sup>T<sub>E</sub>X3 语法环境, 并设置@ 为字母类, 利用 l3cctab 实现。我们使用 \file\_-  
 \ctex\_push\_file: input:n 而不是 L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 的 \input 或者 \InputIfFileExists 载入文件, 因此 L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 的文  
 \ctex\_pop\_file: 件钩子都无效。

```
6070 \cs_new_protected:Npn \ctex_file_input:n #1
6071 {
6072   \ctex_push_file:
6073   \file_input:n {#1}
6074   \ctex_pop_file:
6075 }
6076 \bool_if_exist:NTF \l__kernel_expl_bool
6077 {
6078   \cs_new_protected:Npn \ctex_push_file:
6079   {
6080     \seq_gpush:Nx \g__ctex_expl_status_seq
6081     { \bool_if:NTF \l__kernel_expl_bool { 1 } { 0 } }
6082     \bool_set_false:N \l__kernel_expl_bool
6083     \cctab_begin:N \c__ctex_package_cctab
6084   }
6085   \cs_new_protected:Npn \ctex_pop_file:
6086   {
6087     \cctab_end:
6088     \seq_gpop:NN \g__ctex_expl_status_seq \l__ctex_expl_status_tl
6089     \int_if_odd:nTF { \l__ctex_expl_status_tl }
6090     { \bool_set_true:N \l__kernel_expl_bool }
6091     { \bool_set_false:N \l__kernel_expl_bool }
6092   }
```

```

6093 \tl_new:N \l__ctex_expl_status_tl
6094 \seq_new:N \g__ctex_expl_status_seq
6095 }
6096 {
6097 \cs_new_protected:Npn \ctex_push_file:
6098 { \cctab_begin:N \c__ctex_package_cctab }
6099 \cs_new_protected:Npn \ctex_pop_file:
6100 { \cctab_end: }
6101 }
6102 \cctab_const:Nn \c__ctex_package_cctab
6103 {
6104 \cctab_select:N \c_document_cctab
6105 \char_set_catcode_letter:n { 64 }
6106 }

```

`\ctex_disable_package:n` 禁止宏包载入。采用 L<sup>A</sup>T<sub>E</sub>X 2020-10-01 提供的 `\disable@package@load` 实现, 否则采用传统方式: 预定义 `\ver@{package}.sty` 标识符。

```

6107 \cs_new_protected:Npn \ctex_disable_package:n #1
6108 {
6109 \@@ifpackageloaded {#1}
6110 { \msg_error:nnee }
6111 { \__ctex_disable_package_aux:nnnn }
6112 { ctexhook } { disable-package } {#1} { \@currname }
6113 }
6114 \cs_new_protected:Npn \__ctex_disable_package_aux:nnnn #1#2#3#4
6115 {
6116 \cs_if_exist:NTF \disable@package@load
6117 {
6118 \exp_args:Nne \disable@package@load {#3}
6119 { \msg_warning:nnnn {#1} {#2} {#3} {#4} }
6120 }
6121 { \tl_const:cn { ver@ #3 . \@pkgextension } { 9999/99/99 } }
6122 }
6123 \msg_new:nnn { ctexhook } { disable-package }
6124 { Package~#1'~can~not~be~loaded~with~#2'. }

```

`\ctex_replace_package:nn` 替换宏包。采用 L<sup>A</sup>T<sub>E</sub>X 2020-10-01 提供的 `\declare@file@substitution` 实现, 否则给出无效警告。

```

6125 \ctex_if_format_at_least:nTF { 2020/10/01 }
6126 {
6127 \cs_new_protected:Npn \ctex_replace_package:nn #1#2
6128 {
6129 \declare@file@substitution
6130 { #1 . \@pkgextension }
6131 { #2 . \@pkgextension }
6132 }
6133 }
6134 {
6135 \cs_new_protected:Npn \ctex_replace_package:nn
6136 { \msg_warning:nnnn { ctexhook } { replace-package-invalid } }
6137 \msg_new:nnn { ctexhook } { replace-package-invalid }
6138 {
6139 \token_to_str:N \ctex_replace_package:nn \{#1\}\{#2\}~is~invalid~
6140 before~LaTeX~2020-10-01.
6141 }
6142 }

```

`\ctex_at_begin_package:nn` 如果宏包已经被载入, 则钩子无效, 给出警告。

```

6143 \cs_new_protected:Npn \ctex_at_begin_package:nn #1
6144 {
6145 \@@ifpackageloaded {#1}
6146 { \__ctex_package_loaded_warning:nn {#1} }

```

```

6147     { \ctex_gadd_package_hook:nnn { before } {#1} }
6148   }
6149 \cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
6150   { \msg_warning:nne { ctexhook } { invalid-hook } {#1} }
6151 \msg_new:nnn { ctexhook } { invalid-hook }
6152   {
6153     Package~`#1'~is~loaded. \\
6154     \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
6155   }

```

`\ctex_at_end_package:nn` 与 `filehook` 的 `\AtEndOfPackageFile*` 类似, 如果原来没有在载入宏包则在宏包末尾执行语句, 否则立即执行。

```

6156 \cs_new_protected:Npn \ctex_at_end_package:nn #1
6157   {
6158     \@ifpackageloaded {#1}
6159     { \use:n }
6160     { \ctex_gadd_package_hook:nnn { after } {#1} }
6161   }

```

LaTeX 2020/10/01 开始提供常用钩子管理机制。在新机制下, 我们只需要做简单的包装。

```

6162 \ctex_if_format_at_least:nTF { 2020/10/01 }
6163   {
6164     \cs_new_protected:Npx \ctex_gadd_ltxhook:nn #1
6165     { \hook_gput_code:nnn {#1} { \c_novalue_tl } }
6166     \cs_new_protected:Npn \ctex_at_end_preamble:n
6167     { \ctex_gadd_ltxhook:nn { begindocument/before } }
6168     \cs_new_protected:Npn \ctex_after_end_preamble:n
6169     { \ctex_gadd_ltxhook:nn { begindocument/end } }
6170     \cs_new_protected:Npx \ctex_gadd_package_hook:nnn #1#2
6171     {
6172       \ctex_if_format_at_least:nTF { 2021/11/15 }
6173       { \ctex_gadd_ltxhook:nn { package/#2/#1 } }
6174       { \ctex_gadd_ltxhook:nn { package/#1/#2 } }
6175     }
6176     \file_input_stop:
6177   }
6178 { }

```

对于 LaTeX 2020/10/01 之前的版本, 需要自行补丁。

实现 `etoolbox` 宏包的 `\AtEndPreamble` 和 `\AfterEndPreamble`。

```

\ctex_at_end_preamble:n
\ctex_after_end_preamble:n
6179 \cs_new_protected:Npn \ctex_at_end_preamble:n
6180   { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
6181 \cs_new_protected:Npn \ctex_after_end_preamble:n
6182   { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
6183 \cs_new_protected:Npn \CTEX@document@left@hook
6184   { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
6185 \cs_new_protected:Npn \CTEX@document@right@hook
6186   { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
6187 \cs_set_nopar:Npx \document
6188   {
6189     \CTEX@document@left@hook
6190     \exp_not:o { \document }
6191     \CTEX@document@right@hook
6192   }
6193 \tl_new:N \g__ctex_end_preamble_hook_tl
6194 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

`\ctex_gadd_package_hook:nnn` 给钩子附加内容。

```

\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
6195 \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
6196   { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }

```

```

6197 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1
6198 {
6199   \tl_if_exist:NF #1 { \tl_new:N #1 }
6200   \tl_gput_right:Nn #1
6201 }
6202 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

`\ctex_use_package_hook:nn` 宏包钩子, 只执行一次, 用后清除。

```

6203 \cs_new_protected:Npn \ctex_use_package_hook:nn #1#2
6204 {
6205   \group_begin: \exp_args:Nnc \group_end:
6206   \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
6207 }
6208 \cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
6209 { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }

```

`\@resetoptions` 对 `\@pushfilename` 内部的 `\@currname` 和 `\@currentt` 保存的是前一个宏包的状态, 不能使用。  
`\CTEX@resetoptions@hook` 需要对其后的 `\@resetoptions` 做补丁来实现 `\ctex_at_begin_package:nn` 的功能。

```

6210 \tl_put_right:Nn \@resetoptions { \CTEX@resetoptions@hook }
6211 \cs_new_protected:Npn \CTEX@resetoptions@hook
6212 {
6213   \cs_if_eq:NNT \@currentt \@pkgextension
6214   { \ctex_use_package_hook:nn { before } { \@currname } }
6215 }

```

`\@popfilename` 对 `\@popfilename` 做补丁来实现 `\ctex_at_end_package:nn` 的功能。  
`\CTEX@popfilename@hook`

```

6216 \tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
6217 \cs_new_protected:Npn \CTEX@popfilename@hook
6218 {
6219   \cs_if_eq:NNT \@currentt \@pkgextension
6220   { \ctex_use_package_hook:nn { after } { \@currname } }
6221 }
6222 </ctexhook>

```

## 15.19 ctexpatch 宏包

```
6223 <*ctexpatch>
```

`\ctex_patch_cmd_once:NnnnTF` 只进行第一次匹配进行替换。参数#2 是宏重建时的 `\catcode` 设置。

```

6224 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
6225 {
6226   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1
6227   { once } {#2} { \use_i:nn } { \use_ii:nn }
6228 }

```

`\ctex_patch_cmd_all:NnnnTF` 替换所有匹配到的文本。

```

6229 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
6230 {
6231   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1
6232   { all } {#2} { \use_i:nn } { \use_ii:nn }
6233 }

```

`\ctex_patch_cmd:Nnn` 快捷方式, 在补丁的时候关闭  $\LaTeX$ 3 语法和设置@ 为字母类, 补丁失败时给出警告。

```

6234 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
6235 {
6236   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnw #1

```

```

6237     { once }
6238     {
6239         \ExplSyntaxOff
6240         \char_set_catcode_letter:n { 64 }
6241     }
6242     { }
6243     { \ctex_patch_failure:N #1 }
6244 }
6245 \cs_new_protected:Npn \ctex_patch_failure:N #1
6246 { \msg_warning:nne { ctexpatch } { patch-failure } { \token_to_str:N #1 } }
6247 \msg_new:nnn { ctexpatch } { patch-failure }
6248 { Oops!~Command~#1'~is~NOT~patchable.\\ }

```

`\ctex_preto_cmd:NnnTF` 在宏的原本定义前面增加钩子。

```

6249 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
6250 {
6251     \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
6252     { left } {#2} { \use_i:nn } { \use_ii:nn }
6253 }

```

`\ctex_appto_cmd:NnnTF` 在宏的原本定义后面追加钩子。

```

6254 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
6255 {
6256     \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnnw #1
6257     { right } {#2} { \use_i:nn } { \use_ii:nn }
6258 }

```

`\ctex_patch_boot:NnnTF` 参数记号# 作为宏的参数被读入时，总是会双写，会影响随后的字符串替换。需要先将它转换为普通符号。

```

6259 \cs_new_protected:Npn \ctex_patch_boot:NnnTF #1#2#3#4#5#6
6260 {
6261     \cs_set_protected:Npx \__ctex_patch_true:w { \exp_not:n {#5} }
6262     \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
6263     \group_begin:
6264         \char_set_catcode_other:n { 35 }
6265         \ctex_parse_name:NN #1 #2 {#3} {#4}
6266     }
6267 \cs_new_eq:NN \__ctex_patch_true:w \use_i:nn
6268 \cs_new_eq:NN \__ctex_patch_false:w \use_ii:nn

```

`\ctex_parse_name:NN` 用 `\DeclareRobustCommand` 定义的宏或者由 `\newcommand` 或 `\newrobustcmd` 定义的带一个可选参数的宏第一次展开的结果都不是其实际定义，实际定义被保存在另外的宏中。由这些命令定义的宏的第一次展开结果可以有下面的形式(细节可查阅 `xpatch` 的文档)：

```

1 \protect\xaa_{...} % \DeclareRobustCommand\xaa[1]{...}
2 \protect\xab_{...} % \DeclareRobustCommand\xab[1][...]}
3 \@protected@testopt\yac\yac_{...} % \newcommand\yac[1][...]}
4 \@testopt\xad_{...} % \newrobustcmd\xad[1][...]}
5 \x@protect\1\protect\1_{...} % \DeclareRobustCommand\1[1]{...}
6 \x@protect\2\protect\2_{...} % \DeclareRobustCommand\2[1][...]}
7 \@protected@testopt\3\3_{...} % \newcommand\3[1][...]}
8 \@testopt\4_{...} % \newrobustcmd\4[1][...]}

```

`ctexpatch` 的主要原理是先对宏的 `\meaning` 作字符串替换，然后再用 `\scantokens` 来重建它。我们对宏的实际定义打补丁，为此需要先得到对应的名字。`letltxmacro`、`show2e` 和 `xpatch` 宏包中都有类似的工作。

```

6269 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
6270 { \ctex_parse_name:NNe #1#2 { \cs_to_str:N #2 } }
6271 \group_begin:
6272 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3

```

```

6273 {
6274   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
6275   {
6276     \bool_lazy_or:nnTF
6277     { \cs_if_exist_p:c { ##3 ~ } }
6278     { \cs_if_exist_p:c { #1##3 } }
6279     {
6280       \group_begin:
6281       \use:e
6282       {
6283         \group_end:
6284         \__ctex_parse_name:nNNNnN
6285         { \cs_replacement_spec:N ##2 }
6286         \exp_not:N ##2
6287         \exp_not:c { ##3 ~ }
6288         \exp_not:c { #1##3 }
6289         } { ##3 } ##1
6290       }
6291       { ##1##2 }
6292     }
6293   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
6294   {
6295     \exp_args:Nc ##6
6296     {
6297       \str_case:nnTF {##1}
6298       {
6299         { \protect ##3 } { }
6300         { \x@protect ##2 \protect ##3 } { }
6301       }
6302       {
6303         \str_if_eq:eeTF
6304         { \exp_not:n { #1@protected@ ##3 #1##3 } }
6305         {
6306           \exp_last_unbraced:Ne \__ctex_parse_name:w
6307           { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
6308         }
6309         { #1##5 ~ } { ##5 ~ }
6310       }
6311       {
6312         \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
6313         {
6314           { #1@protected@ ##2 ##4 } { }
6315           { #1@ ##4 } { }
6316         }
6317         { #1##5 } { ##5 }
6318       }
6319     }
6320   }
6321   \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
6322 }
6323 \use:e
6324 {
6325   \__ctex_tmp:w
6326   { \c_backslash_str }
6327   { \c_left_brace_str }
6328   { \tl_to_str:n { testopt } }
6329 }
6330 \group_end:
6331 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNe }

```

\l\_\_ctex\_prefix\_str 分别保存宏的 \meaning 中的前缀、参数文本和替换文本。

\l\_\_ctex\_parameter\_str

\l\_\_ctex\_replacement\_str

```
6332 \str_new:N \l__ctex_prefix_str
```

```
6333 \str_new:N \l__ctex_parameter_str
```

```
6334 \str_new:N \l__ctex_replacement_str
```

`\ctex_get_macro_meaning:NTF` 解构待补丁宏的 `\meaning`。若命令不是宏,则走向false 分支。  
`\__ctex_get_macro_meaning:w`

```

6335 \group_begin:
6336   \cs_set_protected:Npn \__ctex_tmp:w #1
6337     {
6338       \prg_new_protected_conditional:Npnn
6339         \ctex_get_macro_meaning:N ##1 { TF }
6340         {
6341           \exp_after:wN \__ctex_get_macro_meaning:w
6342             \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
6343         }
6344       \cs_new_protected:Npn \__ctex_get_macro_meaning:w
6345         ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
6346         {
6347           \tl_if_empty:nTF { ##4 }
6348             { \prg_return_false: }
6349             {
6350               \str_set:Nn \l__ctex_prefix_str      { ##1 }
6351               \str_set:Nn \l__ctex_parameter_str   { ##2 }
6352               \str_set:Nn \l__ctex_replacement_str { ##3 }
6353               \prg_return_true:
6354             }
6355         }
6356     }
6357   \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
6358 \group_end:

```

`\ctex_if_rescanable:NnTF` 检查宏是否可以重建。

```

6359 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
6360   {
6361     \ctex_get_macro_meaning:NTF #1
6362     {
6363       \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
6364       \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
6365     }
6366     {#4}
6367   }
6368 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

```

`\__ctex_patch_rebuild:Nn` 使用 `\tl_rescan:nn` 来重新记号化 `\meaning` 字符串。

```

6369 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
6370   {
6371     \__ctex_patch_rescan:NNn \l__ctex_prefix_tl      \l__ctex_prefix_str      {#2}
6372     \__ctex_patch_rescan:NNn \l__ctex_parameter_tl   \l__ctex_parameter_str   {#2}
6373     \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
6374     \use:e
6375     {
6376       \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
6377       \exp_not:o { \l__ctex_parameter_tl }
6378       { \exp_not:o { \l__ctex_replacement_tl } }
6379     }
6380   }
6381 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
6382   {
6383     \str_if_empty:NTF #2
6384       { \tl_clear:N #1 }
6385       { \tl_set_rescan:Nno #1 {#3} {#2} }
6386   }
6387 \tl_new:N \l__ctex_prefix_tl
6388 \tl_new:N \l__ctex_parameter_tl
6389 \tl_new:N \l__ctex_replacement_tl

```

`\__ctex_patch_cmd:Nnnnw` 对宏的替换文本进行字符串替换,然后重建。

```

6390 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5

```

```

6391 {
6392   \group_end:
6393   \ctex_if_rescanable:NnTF #1 {#3}
6394   {
6395     \use:e
6396     {
6397       \__ctex_patch_replace:nnnTF {#2}
6398       { \tl_to_str:n {#4} }
6399       { \tl_to_str:n {#5} }
6400     }
6401     {
6402       \__ctex_patch_rebuild:Nn #1 {#3}
6403       \__ctex_patch_true:w
6404     }
6405     { \__ctex_patch_false:w }
6406   }
6407   { \__ctex_patch_false:w }
6408 }

```

`\__ctex_patch_replace:nnnTF` 替换前先检查原文本是否存在。

```

6409 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
6410 {
6411   \tl_if_in:NnTF \l__ctex_replacement_str {#2}
6412   { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
6413 }

```

`\__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

6414 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
6415 {
6416   \group_end:
6417   \ctex_get_macro_meaning:NTF #1
6418   {
6419     \str_if_empty:NTF \l__ctex_parameter_str
6420     { \__ctex_hookto_cmd_parameterless:Nnnnw }
6421     { \__ctex_hookto_cmd_parameter:Nnnnw }
6422     #1 {#2} {#3} {#4}
6423   }
6424   { \__ctex_patch_false:w }
6425 }

```

`\__ctex_hookto_cmd_parameterless:Nnnnw` 如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

6426 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
6427 {
6428   \str_if_empty:NF \l__ctex_prefix_str
6429   { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
6430   \tex_edef:D #1
6431   {
6432     \use:c { __ctex_ #2 _hook_aux:nn }
6433     { \exp_not:o {#1} }
6434     { \exp_not:n {#4} }
6435   }
6436   \__ctex_patch_true:w
6437 }
6438 \cs_generate_variant:Nn \tl_rescan:nn { no }
6439 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
6440 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

`\__ctex_hookto_cmd_parameter:Nnnnw` 如果宏有参数,需要在字符串中进行附加,然后再重建。

```

6441 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
6442 {
6443   \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
6444   \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w

```

```
6445     {
6446       \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
6447       \__ctex_patch_rebuild:Nn #1 {#3}
6448       \__ctex_patch_true:w
6449     }
6450     { \__ctex_patch_false:w }
6451   }

6452 </ctexpatch>
```

# 版本历史

## v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。 . . . . . 44

captiondelimiter 是过时选项。 . . . . . 130

fancyhdr 成为过时选项, 原选项功能总是打开。 . . . . . 47

fntef 成为过时选项, 原选项功能总是打开。 . . . . . 47

hyperref 成为过时选项, 原选项功能总是打开。 . . . . . 47

indent, noindent 是过时选项。 . . . . . 45

nofonts, adobefonts, winfonts 是过时选项。 . . . . . 46

nopunct 是过时选项。 . . . . . 46

nospace 是过时选项。 . . . . . 46

nozhmap 是过时选项。 . . . . . 46

punct 选项可以设置标点格式。 . . . . . 46

ctex 宏包新增 heading 选项。 . . . . . 46

\CTEXindent, \CTEXnoindent 是过时命令。 . . . . . 94

\CTEXsetup, \CTEXoptions 是过时命令。 . . . . . 16, 91

\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; \CTEXfilltwosides 是过时环境。 . . . . . 85

标题设置新增 pagestyle 选项。 . . . . . 98

调整 \footnotesep 的大小, 以适合行距的变化。 . . . . . 129

兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。 . . . . . 48

将标题汉化功能加入 ctex.sty。 . . . . . 95

将中文字号功能提取到可以独立使用的 ctexsize。 . . . . . 125

解决 etoolbox 与 breqn 关于 \end 的冲突。 . . . . . 130

默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。 . . . . . 85

删除 c19gbsn.fd 和 c19gkai.fd。 . . . . . 1

通过 LuaTeX-ja 宏包支持 Lua<sup>®</sup>TeX。 . . . . . 58

新增 autoindent 选项。 . . . . . 45

新增 fontset 选项。 . . . . . 46

新增 linespread 选项。 . . . . . 45

新增 linestretch 选项。 . . . . . 92

新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。 . . . . . 47

新增 zhmCJK 支持选项。 . . . . . 46

新增 zihao 选项。 . . . . . 44

新增统一设置接口 \ctexset。 . . . . . 91

应用 L<sup>®</sup>TeX3 重新整理代码。 . . . . . 1

中文字号不再采用近似值。 . . . . . 125

自动检测操作系统, 载入对应的字体配置。 . . . . . 84

\CJK@surr: 解决与 \nouppercase 的冲突。 . . . . . 52

## v2.0.1 (2015/05/15)

General: 修复 10pt, 11pt 等选项无效的问题。 . . . . . 48

## v2.0.2 (2015/05/16)

General: 修复加载 ctex 宏包后章节标题后第一段无段首缩进的问题。 . . . . . 125

## v2.1 (2015/05/18 – 2015/06/19)

General: format+, nameformat+ 等带加号的选项, 加号与前面的文字之间可以有可选的空格。 . . . . . 97

nameformat 可以接受章节名字为参数。 . . . . . 96

不依赖 ifpdf 宏包。 . . . . . 43

不再设置 hyperref 宏包的 colorlinks 选项。 . . . . . 84

给 pdf<sup>®</sup>TeX 下的非 UTF-8 编码 CJK 字体族加上 CMap。 50

将章节标题设置功能提取到可以独立使用的宏包 ctexheading 中。 . . . . . 1

新的标题格式选项 aftertitle。 . . . . . 97

修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。 . . . . . 119

\ctex\_if\_platform\_macos:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。 . . . . . 131

hyperref: 补充定义 \hypersetup。 . . . . . 47

## v2.2 (2015/06/21 – 2015/06/30)

General: beforeskip 和 afterskip 选项的符号不再有特殊意义。 . . . . . 97

beforeskip, afterskip 和 indent 选项支持表达式。 . 100

不再依赖 etoolbox 宏包。 . . . . . 1

非 ctexart 类的 part/beforeskip 和 part/afterskip 选项有意义。 . . . . . 100

给 enumitem 宏包注册 \chinese 和 \zhnum。 . . . . . 94

将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。 . 157

将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。 . . . . . 119

删去 etoolbox 与 breqn 的兼容补丁。 . . . . . 130

新的标题格式选项 afterindent。 . . . . . 97

新的标题格式选项 numbering。 . . . . . 97

新的标题格式选项 runin。 . . . . . 97

新增子宏包 ctexpatch 实现给宏打补丁功能。 . . . . . 160

## v2.3 (2015/09/17 – 2016/01/05)

General: .value\_required: 和 .value\_forbidden: 已过时。 . . . . . 44

代码实现避免使用 \lowercase 技巧 (Joseph Wright)。 . 43

更新 LuaTeX-ja 支持 (20150922.0)。 . . . . . 58

更新 unicode-math 宏包补丁。 . . . . . 61

兼容 titles 宏包。 . . . . . 112

修复 nameformat 作用域问题。 . . . . . 96

与 L<sup>®</sup>TeX3 (2015/12/20) 同步。 . . . . . 41

## v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 up<sup>®</sup>TeX。 . . . . . 81

加强 beamer 宏包支持。 . . . . . 1

提供 up<sup>®</sup>TeX 的 NFSS 字体定义。 . . . . . 133

提供 translator 宏包的中文定理名称翻译。 . . . . . 155

正确更新 CJK 包的 \CJKfamilydefault。 . . . . . 83

正确设置 up<sup>®</sup>TeX 下字体命令。 . . . . . 81

\ctex\_parse\_name:NN: 修复宏名解析错误。 . . . . . 161

## v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。 . . . . . 124

beamer 不调整默认字体大小。 . . . . . 124

使用 bootfont.bin 判断 Windows XP 以避免权限问题。 . . . . . 145

随字体更新 up<sup>®</sup>TeX 的 \xkanjiskip。 . . . . . 88

新的标题格式选项 part/fixbeforeskip 和 chapter/fixbeforeskip。 . . . . . 98

正确更新 up<sup>®</sup>TeX 的 \CJKfamilydefault。 . . . . . 83

\ccwd: 正确设置 up<sup>®</sup>TeX 下的 \ccwd。 . . . . . 86

\chinese: 支持 \pagenumbering。 . . . . . 94

zihao: 不允许无参 zihao 选项。 . . . . . 45

<b>v2.4.2</b>	(2016/05/15)	<b>v2.4.13</b>	(2018/03/23)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。 . . . . .	61	<code>\CTEX@selectfont@hook</code> : 修正导言区 <code>\selectfont</code> 钩子位置。 . . . . .	87
<code>\em</code> : 兼容 <code>upL<sup>A</sup>T<sub>E</sub>X 2016/05/07u00</code> 的定义。 . . . . .	81	<b>v2.4.14</b>	(2018/05/01)
<b>v2.4.3</b>	(2016/06/03 – 2016/08/26)	General: 配置 <code>macnew</code> 的默认字体设置。 . . . . .	140
General: 更新 <code>unicode-math</code> 补丁。 . . . . .	67	区分 <code>macold</code> 及 <code>macnew</code> 。 . . . . .	139
简化 <code>fontspec</code> 补丁。 . . . . .	61	为 <code>macnew</code> 配置字体命令。 . . . . .	146
确保 <code>\proofname</code> 非空。 . . . . .	95	<b>v2.4.15</b>	(2019/01/29 – 2019/04/05)
删除选项 <code>part/fixbeforeSKIP</code> 和 <code>chapter/fixbeforeSKIP</code> 。 . . . . .	98	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。 . . . . .	100
新的标题格式选项 <code>fixSKIP</code> 。 . . . . .	97	将 <code>upL<sup>A</sup>T<sub>E</sub>X</code> 的默认字体由 <code>mc</code> 改为 <code>zhrm</code> , 并启用 <code>\jfam</code> 。 . . . . .	81
<b>v2.4.4</b>	(2016/09/09 – 2016/09/19)	将 <code>JY2</code> 和 <code>JT2</code> 编码的字体定义提取到单独的文件中。 . . . . .	132
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。 . . . . .	58	局部指定 <code>autoindent</code> 为 <code>false</code> , 并交换 <code>\CTEX@XXX@indent</code> 与 <code>\CTEX@XXX@format</code> 的顺序。 . . . . .	115
改进 <code>hyperref</code> 宏包的标题锚点设置。 . . . . .	108	同步 <code>L<sup>A</sup>T<sub>E</sub>X 3 2019/03/05</code> 。 . . . . .	1
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。 . . . . .	52	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。 . . . . .	137
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT<sub>E</sub>X</code> 的编号。 . . . . .	111	显式补丁 <code>upL<sup>A</sup>T<sub>E</sub>X</code> 的 <code>\rmfamily</code> 等字体命令。 . . . . .	81
提供 <code>\CTEXifname</code> 。 . . . . .	99	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。 . . . . .	100
提供 <code>\partmark</code> 。 . . . . .	99	<b>v2.4.16</b>	(2019/05/11 – 2019/05/29)
新的标题格式选项 <code>break</code> 。 . . . . .	97	General: 更好地兼容 <code>nameref</code> 宏包。 . . . . .	109
<b>v2.4.5</b>	(2016/10/01 – 2016/10/25)	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法, 在其标题内部禁用 <code>autoindent</code> 。 . . . . .	100
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。 . . . . .	98	允许设置 <code>autoindent</code> 为 0。 . . . . .	43
新的标题格式选项 <code>hang</code> 。 . . . . .	97	<b>v2.5</b>	(2019/10/25 – 2020/04/30)
新的标题格式选项 <code>tocline</code> 。 . . . . .	97	General: <code>ubuntu</code> 改用思源(Noto CJK)和文鼎字库, 不再支持使用 <code>pdfL<sup>A</sup>T<sub>E</sub>X</code> 编译。 . . . . .	144
<code>\ps@headings</code> : 修复补丁失败。 . . . . .	114	标题选项 <code>format</code> 也可以接受参数。 . . . . .	100
<b>v2.4.6</b>	(2016/10/31 – 2016/11/20)	不再支持 Windows XP 系统, <code>windowsold</code> 和 <code>windowsnew</code> 成为过时字库选项。 . . . . .	145
General: <code>\CTEXifname</code> 初始为假。 . . . . .	99	不再自动载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。 . . . . .	85
支持字体属性可选项在后的新语法。 . . . . .	72	操作系统检测移动至载入中文字库处, 且不再需要依赖特定引擎。 . . . . .	84
重新初始化 <code>\ifthechapter</code> 等。 . . . . .	113	处理 <code>\ctex_file_input:n</code> 在 <code>ctexsize</code> 中未定义的错误。 . . . . .	41
<b>v2.4.7</b>	(2016/12/23 – 2016/12/27)	给 <code>L<sup>A</sup>T<sub>E</sub>X</code> 和 <code>upL<sup>A</sup>T<sub>E</sub>X</code> 下的文档类指定驱动为 <code>DVIPDFMx</code> 。 . . . . .	49
General: 依赖 <code>pxeverysel</code> 宏包。 . . . . .	82	更新 <code>LuaTeX-ja</code> 支持(20200412.0)。 . . . . .	58
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。 . . . . .	114	兼容 <code>titletoc</code> 宏包。 . . . . .	113
<b>v2.4.8</b>	(2017/02/23)	删除 <code>fontspec</code> 补丁。 . . . . .	61
General: 解决与 <code>fontspec 2017/01/24 v2.5d</code> 的字体族匹配兼容问题。 . . . . .	74	使用环境变量代替绝对路径查找字体。 . . . . .	145
<b>v2.4.9</b>	(2017/02/27)	所有引擎下默认编码均设为 UTF-8。 . . . . .	45
General: 调整 <code>unicode-math</code> 补丁的代码顺序。 . . . . .	67	为 <code>macnew</code> 增加粗楷体、隶书和圆体的定义。 . . . . .	140
<b>v2.4.10</b>	(2017/07/19 – 2017/07/23)	移除 <code>\CTEXunderdot</code> 、 <code>\CTEXunderline</code> 、 <code>\CTEXunderdblline</code> 、 <code>\CTEXunderwave</code> 、 <code>\CTEXsout</code> 、 <code>\CTEXxout</code> 、 <code>CTEXfilltwosides</code> 等命令和环境。 . . . . .	85
General: 常数 <code>\c_minus_one</code> 已过时。 . . . . .	41	应用新内核中的 <code>\labelformat</code> 。 . . . . .	118
定义 <code>\cht</code> 、 <code>\cdp</code> 和 <code>\c wd</code> 。 . . . . .	89	允许 <code>macnew</code> 在 <code>L<sup>A</sup>T<sub>E</sub>X</code> 和 <code>upL<sup>A</sup>T<sub>E</sub>X</code> 下使用。 . . . . .	140
使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 ( <code>L<sup>A</sup>T<sub>E</sub>X 3 2017/07/19</code> )。 . . . . .	41	在 <code>ctexsize</code> 也载入 <code>fix-cm</code> 。 . . . . .	41
<b>v2.4.11</b>	(2017/08/17 – 2017/11/21)	增加宏包开头钩子。 . . . . .	157
General: 不把 Enclosed Alphanumerics 设置为 <code>JAchar</code> 。 . . . . .	59	增加字体映射文件 <code>zhmacfonts.tex</code> 。 . . . . .	150
不把希腊和西里尔字母设置为 <code>JAchar</code> 。 . . . . .	59	重构标题选项 <code>indent</code> 和 <code>hang</code> 。 . . . . .	98
因上游 <code>l3keys</code> 变化, 重新定义 <code>format<sub>l</sub>+<sub>+</sub></code> 等带空格加号的选项。 . . . . .	97	重构字体选项 <code>AlternateFont</code> 。 . . . . .	74
<code>\ps@headings</code> : 补充页眉空格。 . . . . .	114	<code>\CTEX@selectfont@hook</code> : 兼容 KOMA-Script 的 <code>\selectfont</code> 补丁。 . . . . .	87
<b>v2.4.12</b>	(2017/12/05 – 2018/01/27)	<code>\ctex_if_platform_macos:TF</code> : 改用 <code>/System/Library/Fonts/Menlo.ttc</code> 为特征文件。 . . . . .	131
General: 同步 <code>L<sup>A</sup>T<sub>E</sub>X 3 2017/12/16</code> 。 . . . . .	1	<code>fntef</code> : 仅在该选项启用时会载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。 . . . . .	47
修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。 . . . . .	91		
正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。 . . . . .	80		
<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。 . . . . .	129		

<b>v2.5.1</b>	(2020/05/02)	<b>v2.5.10</b>	(2022/06/10 – 2022/07/11)
General: zhconv 更名为 ctex-zhconv。 . . . . .	1	General: 不直接依赖 xparse 和 l3keys2e。 . . . . .	41
<b>v2.5.2</b>	(2020/05/05 – 2020/05/06)	更新一些内部函数。 . . . . .	41
General: ctexmakespa.tex 更名为 ctex-spa-make.tex。 152		解决 CJK 包与 \text_uppercase:n 等转化函数的冲突。 . . . . .	54
ctexspamacro.tex 更名为 ctex-spa-macro.tex。 . . . . .	152	取消 L <sup>A</sup> T <sub>E</sub> X 2022-06-01 对书名号的定义。 . . . . .	52
zhadobefonts.tex 等字体映射文件更名为		展开传递 pagestyle 的值。 . . . . .	99
ctex-zhmap-*.tex。 . . . . .	147	\ctex_add_cmap:n: 使用封装好的函数。 . . . . .	51
兼容 L <sup>A</sup> T <sub>E</sub> X 2020-02-02 之前的版本。 . . . . .	82	<b>v2.6.0</b>	(2024/04/20 – 2026/06/23)
新增标题选项 secnumdepth 和 tocdepth。 . . . . .	116	General: macnew 增加 macOS 15+ 兼容, 字体运行时检测。 139	
修正 macnew 和 ubuntu 字库的 CJKpunct 标点信息。 . . . . .	139	<b>Breaking:</b> \newCJKfontfamily 定义的字体切换命令改为局部定义, 与 \newcommand 行为一致。在分组内调用时, 命令不再泄漏到分组外(#751)。 . . . . .	72
<b>v2.5.3</b>	(2020/05/31 – 2020/06/06)	<b>Breaking:</b> 移除对 L <sup>A</sup> T <sub>E</sub> X 2020/10/01 之前版本的字体钩子兼容代码。现在需要 L <sup>A</sup> T <sub>E</sub> X 2020/10/01 或更新版本(#746)。 . . . . .	82
General: 不再依赖 xunicode, 单独补丁 tuenc.def。 . . . . .	60	4 字节 UTF-8 字符也优先使用	
兼容 cleveref。 . . . . .	117	\DeclareUnicodeCharacter 的定义(#815)。 . . . . .	54
正确关闭和恢复 L <sup>A</sup> T <sub>E</sub> X3 语法环境。 . . . . .	42	补全 upL <sup>A</sup> T <sub>E</sub> X 字体编码 JY2 和 JT2 的 Fallback 机制。 . . . . .	133
<b>v2.5.4</b>	(2020/06/07 – 2020/08/16)	将 experiment/font-size-system 的 traditional 选项更名为 letterpress(#813)。 . . . . .	44
General: 更新 LuaTeX-ja 支持(20200808.0)。 . . . . .	58	使用 \l_keys_key_str 和 \l_keys_choice_str 替代已废弃的 \tl 版本(#806)。 . . . . .	44
兼容 L <sup>A</sup> T <sub>E</sub> X 2020/10/01 的钩子机制。 . . . . .	157	提升 L <sup>A</sup> T <sub>E</sub> X3 版本至 2022/10/09。 . . . . .	41
同时兼容 cleveref 和 hyperref。 . . . . .	117	提升 L <sup>A</sup> T <sub>E</sub> X3 最低版本要求至 2025/10/09。 . . . . .	41
修正主要字体命令补丁。 . . . . .	82	文档: 在标准字体命令、中文字号表附近提示	
应用 l3cctab。 . . . . .	42	experiment/font-size-system 的影响; 说明	
<b>v2.5.5</b>	(2020/10/06 – 2020/10/17)	letterpress 只是金属活字排印字号体系之一(#871)。 . . . . .	44
General: 放弃应用 l3cctab。 . . . . .	42	文档说明 runin 与 aftertitle 的交互(#574)。 . . . . .	23
进一步应用 L <sup>A</sup> T <sub>E</sub> X 2020/10/01 的新钩子。 . . . . .	83	文档字体统一为 Noto CJK 系列(#686)。 . . . . .	1
\CJKhook: 不再通过旧的钩子命令来定义。 . . . . .	57	新增实验性 experiment/CJKecglue 选项(#717)。 . . . . .	90
<b>v2.5.6</b>	(2021/01/11 – 2021/02/16)	新增实验性 experiment/font-size-system 选项(#543)。 . . . . .	44
General: 使用正确的导言区末尾钩子。 . . . . .	159	修复 hyperref 在 ctex 之前被加载时 driverfallback 选项重复设置的警告(#715)。 . . . . .	84
\f@nch@initialise: 更新 fancyhdr 宏包的补丁。 . . . . .	114	修复参数签名, \_ctex_char_auxi:NNNN 原误为 NNN。 . . . . .	54
<b>v2.5.7</b>	(2021/06/04 – 2021/06/12)	修复了 T <sub>E</sub> X tree 字体无法制作 .spa 文件的问题。 . . . . .	152
General: 更好地兼容 cmap 包。 . . . . .	51	修复内联 \verb 前 xkanjiskip 丢失的问题(移植	
兼容 L <sup>A</sup> T <sub>E</sub> X 2021/06/01 的字体钩子。 . . . . .	41	lltjcore 对 \verb 和 \do@noligs 的修复)。 . . . . .	80
禁用 DVIPDFMx 驱动的 unicode 书签设置。 . . . . .	84	\ctex_undeclare_unicode_character:n: 替换废弃命令	
使用 \disable@package@load 禁止宏包载入。 . . . . .	158	\char_to_utfviii_bytes:n。 . . . . .	52
同时兼容 cleveref 和 beamer。 . . . . .	117	<b>v2.6.1</b>	(2026/07/01)
重新应用 l3cctab。 . . . . .	157	General: 补全字库选项 windows 的 pdfL <sup>A</sup> T <sub>E</sub> X 分支中,	
\CJK@addcmap: 应用 \pdfnobluiltintounicode。 . . . . .	53	\ctex_zhmap_case:nnn 的第三个子分支 zhmap =	
\ctex_add_cmap:n: 确保 cmap 文件存在。 . . . . .	51	false 缺失的报错。 . . . . .	145
<b>v2.5.8</b>	(2021/11/18 – 2021/12/04)		
General: 兼容 L <sup>A</sup> T <sub>E</sub> X 2021/11/15。 . . . . .	159		
兼容 microtype。 . . . . .	63		
兼容 titlesec 包和 \CTEXifname。 . . . . .	111		
简化部分 Lua 函数。 . . . . .	61		
<b>v2.5.9</b>	(2022/05/26 – 2022/05/27)		
General: 设置消息模块的名字和类型。 . . . . .	41		
依赖 chinese-jfm 宏包。 . . . . .	68		

# 代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

<b>Symbols</b>	
\#	3517, 3576
.../afterindent	26
.../aftername	22
.../aftername+	22
.../afterskip	25
.../aftertitle	22
.../aftertitle+	22
.../beforeskip	25
.../break	26
.../break+	26
.../fixskip	25
.../format	20
.../format+	20
.../hang	24
.../indent	25
.../name	18
.../nameformat	20
.../nameformat+	20
.../number	19
.../numberformat	21
.../numberformat+	21
.../titleformat	22
.../titleformat+	22
.../tocline	27
\/	1911
\\	17, 18, 91, 147, 149, 201, 210, 830, 1396, 1478, 2617, 4106, 4107, 4219, 4220, 4326, 4540, 4542, 4571, 6058, 6059, 6060, 6153, 6248
\{	1407, 1408, 6060, 6139, 6154
\}	1407, 1408, 6060, 6139, 6154
\_	3623, 3631, 3644, 3647, 3673, 3681
10pt	10, <u>329</u>
11pt	10, <u>329</u>
12pt	10, <u>329</u>
<b>Numbers</b>	
\1	5027
<b>A</b>	
\abovedisplayshortskip	4382, 4390, 4401, 4421, 4429, 4440
\abovedisplayskip	4381, 4384, 4389, 4396, 4400, 4407, 4420, 4423, 4428, 4435, 4439, 4446
\abstractname	2601
abstractname	14
\addCJKfontfeature	1478, 1529
\addCJKfontfeatures	1495
\addcontentsline	2831, 2876, 2880, 2885, 2917, 2921, 2926, 3026, 3031, 3037, 3043, 3172, 3186
\AddEnumerateCounter	2556, 2558, 2559, 2560
\addpenalty	3129, 3947, 3980, 4014, 4028, 4042
\addtocontents	2819, 2825, 3048, 3049
\addvspace	2820, 2826, 2837, 2840, 3048, 3049, 3129, 3133, 3697, 3708, 3716, 3727, 3735, 3746
adobefonts	36
\algorithmname	2608, 2624
algorithmname	14
AlternateFont	32, 38, <u>1614</u>
\appendix	<u>3341</u> , 3823
appendix/name	28, <u>3330</u>
appendix/number	28, <u>3330</u>
appendix/numbering	27, <u>3330</u>
\appendixname	2603, 4091
appendixname	14
\arabic	3994
\AtBeginDocument	570, 1888
\AtBeginDvi	434
\AtBeginShipoutFirst	439
\AtBeginUTFCommand	960
\AtEndOfClass	115
\AtEndOfPackage	116
\AtEndUTFCommand	961
autoindent	12, <u>180</u> , <u>2402</u>
<b>B</b>	
\baselinestretch	2461
\begin	3704, 3723, 3742
\begingroup	400, 2890, 2932, 2952, 2973, 3072, 3097, 3162, 3200, 3233, 3692, 3711, 3730, 5485, 5519, 5526
\belowdisplayshortskip	4383, 4391, 4402, 4422, 4430, 4441
\belowdisplayskip	4384, 4396, 4407, 4423, 4435, 4446
\bfdefault	1263, 1269, 1604, 1611
\bfseries	2895, 2899, 2937, 2941, 2957, 2978, 3078, 3083, 3102, 3875, 3877, 3881, 3894, 3895, 3898, 3924, 3926, 3932, 3949, 3953, 3974, 4008, 4025, 4039
\bgroup	1903
\bibname	2606, 2609, 2617, 2625, 2631, 2632
bibname	14
<b>bool commands:</b>	
\bool_case:nTF	5861
\bool_const:Nn	49, 52
\bool_gset_false:N	243, 248
\bool_gset_true:N	238, 5033, 5174, 5280
\bool_if:NTF	504, 579, 1436, 1989, 2208, 3255, 3447, 3457, 3468, 3811, 4110, 5023, 5471, 6081
\bool_if_exist:NTF	6076
\bool_lazy_and:nnTF	5874
\bool_lazy_or:nnTF	4517, 5463, 6276
\bool_new:N	111, 1473, 5016, 5150
\bool_set_false:N	1448, 6082, 6091
\bool_set_true:N	1466, 6090

\c_false_bool .....	49, 3289, 3291	4664, 4668, 4669, 4671, 4672, 4674, 4675, 4703, 4704,
\c_true_bool .....	52, 3288, 3290	4706, 4707, 4709, 4710, 4714, 4715, 4717, 4718, 4720, 4721
<b>box commands:</b>		\CJKfamily . . . . 1335, <u>1355</u> , 1709, 1919, 2014, 2015, 2016,
\box_ht:N .....	4462	2017, 5453, 5454, 5455, 5456, 5458, 5459, 5461, 5473, 5474
\box_new:N .....	57	\CJKfamilydefault .....
\box_use_drop:N .....	3265	587, 1568, 1570, 1574, 1581,
\box_wd:N .....	2145, 3264, 3572	1594, 2008, 2009, 2017, 2026, 2033, 2037, 2045, 2056, 2058
		\CJKfontspec .....
		1495
		\CJKglue .....
		2144, 2158, 2161, 2174, 2186
		\CJKhook .....
		<u>761</u>
		\CJKnormal . . . . 4610, 4611, 4612, 4613, 4614, 4615, 4616,
		4621, 4622, 4623, 4624, 4625, 4626, 4627, 4656, 4659,
		4662, 4667, 4670, 4673, 4702, 4705, 4708, 4713, 4716, 4719
		\CJKpunctmapfamily .....
		. 796, 797, 798, 799, 800, 801, 805, 806, 807, 808, 809, 810
		\CJKrmdefault .....
		. 518, 1403, 1414, 1533, 1570, 1573, 1574, 2009, 2011,
		2014, 2020, 2023, 2041, 4772, 4775, 4776, 4781, 4782,
		4783, 4836, 4839, 4841, 4846, 4847, 4848, 4902, 4905,
		4906, 4913, 4914, 4915, 4967, 4970, 4972, 4977, 4978,
		4979, 5103, 5106, 5107, 5114, 5116, 5117, 5336, 5339,
		5342, 5348, 5349, 5350, 5405, 5406, 5407, 5415, 5416, 5417
		\CJKsfdefault .....
		519, 1404, 1415, 1540,
		2012, 2015, 2021, 2024, 2042, 4773, 4837, 4840, 4903,
		4907, 4968, 4971, 5104, 5108, 5115, 5337, 5340, 5408, 5409
		\CJKtilde .....
		588
		\CJKttdefault . . . 520, 1405, 1416, 1546, 2013, 2016, 2022,
		2025, 2043, 4774, 4838, 4904, 4969, 5105, 5338, 5342, 5410
		clearalternatefont .....
		33, <u>1793</u>
		\cleardoublepage .....
		2851, 3004, 3891, 3919
		\clearpage .....
		2853, 3004, 3891, 3919
		<b>clist commands:</b>
		\clist_clear:N .....
		1284, 5844
		\clist_concat:NNN .....
		1304
		\clist_const:Nn .....
		775
		\clist_gput_right:Nn .....
		345, 357, 383, 384
		\clist_gset:Nn .....
		1556
		\clist_if_empty:NTF .....
		1639
		\clist_map_break:n .....
		3426, 4101
		\clist_map_function:nN . . . 1622, 1796, 1798, 1831, 1845
		\clist_map_inline:Nn .....
		765, 1739
		\clist_map_inline:n .....
		330,
		788, 1838, 2136, 3421, 4098, 4242, 4265, 4296, 4344, 5831
		\clist_new:N . . . . 126, 1301, 1557, 1635, 1682, 5889, 6053
		\clist_put_left:Nn .....
		1305, 4167, 4192
		\clist_put_right:Nn .....
		1465, 1728, 5883
		\clist_set:Nn . . . . 1288, 1625, 1665, 1666, 1885, 6054
		\clist_use:Nn .....
		6063
		\clubpenalty .....
		3232, 3239
		<b>codepoint commands:</b>
		\codepoint_str_generate:n .....
		526
		\contentsname .....
		2596
		contentsname .....
		13
		continuation .....
		14
		<b>cs commands:</b>
		\cs_generate_variant:Nn .....
		. . . . . 80, 458, 1236, 1367, 1454, 1474, 1475, 1683,
c5size .....	35	
cap .....	36, <u>288</u>	
captiondelimiter .....	37, <u>4469</u>	
\catcode .....	5485, 5486, 5487, 5488, 5489, 5493, 5494	
CCT .....	35	
<b>cctab commands:</b>		
\cctab_begin:N .....	6083, 6098	
\cctab_const:Nn .....	6102	
\cctab_end: .....	6087, 6100	
\cctab_select:N .....	6104	
\c_document_cctab .....	6104	
CCTfont .....	35	
\ccwd .....	31, 109, 184, <u>2141</u> , 2199,	
	2200, 2201, 2408, 2446, 2472, 2473, 2480, 2506, 2507,	
	2508, 2511, 2513, 2514, 2525, 2536, 2538, 4479, 4480, 4481	
\cdp .....	<u>2296</u>	
\centering .....	2933, 2974, 3693,	
	3712, 3731, 3881, 3893, 3898, 3902, 3932, 3953, 3961, 3996	
\chapter .....	<u>3003</u> , 4116	
chapter/beforeskip .....	37	
chapter/fixbeforeskip .....	37	
chapter/lofskip .....	27, <u>2742</u>	
chapter/lotskip .....	27, <u>2742</u>	
chapter/numbering .....	17	
chapter/pagestyle .....	23, <u>2742</u>	
\chaptermark .....	3047	
\chaptername .....	3921	
<b>char commands:</b>		
\char_generate:n .....		
	727, 730, 731, 735, 736, 737, 741, 742, 743, 744	
\char_set_catcode:n .....	679	
\char_set_catcode_active:n .....	672	
\char_set_catcode_comment:n .....	848	
\char_set_catcode_ignore:n .....	5057, 5248	
\char_set_catcode_letter:n .....	1891, 3592, 3831, 6105, 6240	
\char_set_catcode_other:N .....	1163, 3517, 3576	
\char_set_catcode_other:n .....	6264	
\char_set_catcode_space:n .....	966, 2301, 5040, 5200	
\char_value_catcode:n .....	681	
CharRange .....	32, <u>1614</u>	
\Chinese .....	37, 2552	
\chinese .....	31, <u>2550</u> , 2560, 3868, 3931	
\cht .....	<u>2296</u>	
\CJKaddEncHook .....	527	
\CJKbold .....		
	4617, 4618, 4628, 4629, 4657, 4658, 4660, 4661, 4663,	

1699, 1771, 1792, 1831, 1839, 3267, 3768, 6202, 6331, 6438	\cs_new_protected:Npx . . . . . 480, 919, 3494, 6164, 6170
\cs_gset:Npn . . . . . 650, 657, 664	\cs_new_protected_nopar:Npn . . . . . 1082, 1096, 1098
\cs_gset_eq:NN . . . . . 239, 244, 249, 441, 500, 501,	\cs_replacement_spec:N . . . . . 6285, 6307
643, 644, 645, 646, 675, 1441, 1444, 3378, 3379, 3385, 3410	\cs_set:Npn . . . . . 2213, 3557
\cs_gset_protected:Npn . . . . . 493, 3342, 3391, 3406	\cs_set_eq:NN . . . . . 98, 185, 190, 195, 526, 539,
\cs_gset_protected:Npx . . . . . 1332, 1703, 1789, 3318	559, 756, 760, 769, 784, 900, 933, 941, 948, 950, 951,
\cs_gset_protected_nopar:Npx . . . . . 1106	1050, 1075, 1076, 1175, 1205, 1223, 1278, 1335, 1702,
\cs_if_eq:NNTF . . . . . 1197, 3376, 3409, 6213, 6219, 6364, 6444	1709, 1719, 1720, 1721, 1787, 1788, 1911, 2036, 2161,
\cs_if_exist:NTF . . . . . 3, 24,	2231, 2234, 2235, 2238, 2567, 2572, 2575, 2580, 2810,
321, 363, 426, 469, 1056, 1072, 1104, 1138, 1216, 1507,	2812, 3279, 3280, 3281, 3282, 3283, 3284, 3373, 3417, 3427
1702, 1787, 2160, 2260, 3492, 3788, 3838, 4116, 4118, 6116	\cs_set_nopar:Npn . . . . . 1937
\cs_if_exist_p:N . . . . . 6277, 6278	\cs_set_nopar:Npx . . . . . 3455, 6187
\cs_if_exist_use:N . . . . . 1351, 1352	\cs_set_protected:Npn . . . . .
\cs_if_exist_use:NTF . . . . .	755, 1209, 1224, 2158, 2190, 3487, 3555, 4371, 6272, 6336
. . . . . 652, 659, 666, 916, 1038, 1373, 1807, 6209	\cs_set_protected:Npx . . . . . 911, 1511, 2204, 6261, 6262
\cs_if_free:NTF . . . . .	\cs_to_str:N . . . . . 67, 1506, 1997, 2002, 6270
. . . . . 452, 462, 540, 561, 767, 783, 1438, 1603, 2210,	\cs_undefine:N 52, 1315, 1316, 1320, 1321, 1722, 2239, 6209
2219, 2222, 2421, 2435, 2556, 3440, 3485, 3542, 3547, 3551	cs4size . . . . . 35
\cs_new:Npn . . . . . 101, 108, 592, 594, 600, 608,	\csname . . . . . 401, 404, 405, 3130, 3139,
614, 616, 625, 631, 633, 673, 710, 712, 714, 726, 728,	3140, 3141, 3148, 3168, 3169, 3171, 3174, 3183, 3184,
733, 739, 746, 747, 748, 1123, 1153, 1158, 1160, 1164,	3185, 3188, 3833, 3834, 5518, 5520, 5521, 5522, 5527, 5531
1280, 1346, 1354, 1399, 1410, 1860, 2550, 2724, 3292,	\CTeX . . . . . 31, <u>4465</u>
3300, 3308, 3327, 3465, 3540, 3545, 5890, 6068, 6321, 6439	ctex commands:
\cs_new:Npx . . . . . 4488, 4494	\ctex_add_cmap:n . . . . . 455, <u>460</u> , 460, 500
\cs_new_eq:NN . . . . . 95,	\ctex_add_dvi_zhmap: . . . . . 433, 435, 441
96, 97, 113, 114, 322, 459, 487, 591, 814, 1081, 1108,	\ctex_add_to_selectfont:n . . . . .
1174, 1204, 1529, 1536, 2052, 2186, 2207, 2216, 2551,	. . . . . <u>2202</u> , 2202, 2266, 2294, 2318, 2434
2552, 2553, 2782, 2801, 2808, 3276, 3286, 3287, 3288,	\ctex_after_end_preamble:n 43, 98, 435, 6168, <u>6179</u> , 6181
3289, 3290, 3291, 3341, 3365, 3366, 3841, 4114, 4119,	\ctex_appto_cmd:NnnTF . . . . . 938, 1890, <u>6254</u> , 6254
4234, 5467, 5468, 5477, 5478, 5836, 6267, 6268, 6368, 6440	\ctex_assign_heading_name:nn . . 2694, <u>2732</u> , 2732, 3334
\cs_new_protected:Npn . . . . .	\ctex_at_begin_package:nn . 160, 836, <u>6143</u> , 6143, 6154
. . . 60, 63, 65, 67, 69, 99, 115, 116, 118, 134, 136, 141,	\ctex_at_end:n . . . . . <u>115</u> ,
422, 424, 428, 433, 438, 450, 460, 465, 467, 472, 516,	115, 116, 261, 297, 303, 314, 319, 677, 848, 2186, 2188, 4578
525, 532, 542, 683, 689, 695, 701, 707, 708, 709, 753,	\ctex_at_end_package:nn . . . . .
759, 763, 781, 786, 794, 803, 812, 888, 890, 896, 898,	. . . . . 160, 436, 498, 838, 957, 963, 1214, 2079, 2082,
906, 908, 913, 926, 931, 946, 1035, 1045, 1059, 1064,	2108, 2116, 2123, 2554, 3389, 3404, 3473, 3480, 3505,
1070, 1125, 1136, 1148, 1167, 1176, 1192, 1206, 1231,	3550, 3552, 3553, 3589, 3635, 3797, 3847, <u>4467</u> , <u>6156</u> , 6156
1281, 1302, 1311, 1330, 1347, 1357, 1382, 1420, 1432,	\ctex_at_end_preamble:n . . . . .
1434, 1456, 1458, 1481, 1564, 1589, 1598, 1621, 1623,	. . . . . 324, 578, 584, 2030, 6166, <u>6179</u> , 6179
1630, 1637, 1643, 1650, 1652, 1657, 1663, 1684, 1693,	\ctex_at_shipout_first:n 428, 430, 438, 440, 2086, 2110
1700, 1706, 1717, 1725, 1731, 1737, 1755, 1764, 1772,	\ctex_auto_ignorespaces: . . . . . <u>591</u> , 591, 760, 2328
1784, 1802, 1819, 1837, 1840, 1842, 1855, 1874, 1882,	\ctex_CJK_input:n . . . . . <u>532</u> , 532, 539
1896, 1912, 1914, 1938, 1952, 1965, 1978, 1994, 1999,	\ctex_cleveref_hook: . . . . . 3797, <u>3809</u> , 3809
2031, 2063, 2067, 2119, 2127, 2131, 2141, 2155, 2198,	\ctex_declare_math_sizes:nnnn . . . . . <u>4329</u> , <u>4329</u> , 4363
2202, 2217, 2251, 2255, 2275, 2420, 2422, 2447, 2455,	\ctex_define:n . . . . . <u>59</u> , 63,
2467, 2469, 2488, 2498, 2503, 2658, 2688, 2735, 2765,	1793, 1832, 2320, 2341, 2353, 2402, 2436, 2568, 2594,
2772, 2777, 2785, 2792, 2799, 2803, 2809, 2811, 2814,	2622, 2632, 2635, 2638, 2764, 3330, 3753, 3804, 4469, 4545
2830, 3244, 3250, 3252, 3260, 3268, 3277, 3358, 3368,	\ctex_define_option:n . . . . . <u>59</u> , 60, 156, 351, 354
3402, 3413, 3433, 3438, 3443, 3453, 3463, 3513, 3518,	\ctex_deprecated_command:Nn . . . . .
3566, 3584, 3762, 3784, 3793, 3799, 3809, 3826, 4210,	. . . . . <u>134</u> , 141, 2385, 2387, 2395, 2397, 2535, 2542
4225, 4329, 4334, 4364, 4370, 4484, 4500, 4510, 4512,	\ctex_deprecated_option:n . . . . .
5828, 5837, 5881, 6070, 6078, 6085, 6097, 6099, 6107,	. . . . . 134, 138, 199, 208, 296, 302, 306, 313, 318, 4473
6114, 6127, 6135, 6143, 6149, 6156, 6166, 6168, 6179,	\ctex_deprecated_option:nn . . . . . <u>134</u> , 263
6181, 6183, 6185, 6195, 6197, 6203, 6208, 6211, 6217,	\ctex_detect_platform: . . . . . <u>4500</u> , 4500, 4515, 4530
6224, 6229, 6234, 6245, 6249, 6254, 6259, 6269, 6274,	\ctex_disable_package:n . 40, 41, 43, 44, 840, <u>6107</u> , 6107
6293, 6344, 6359, 6369, 6381, 6390, 6409, 6414, 6426, 6441	\ctex_family_cmap:nn . . . . . <u>450</u> , 450, 458, 459

- \ctex\_file\_input:n ..... 59, 360, 521, 815, 949,  
2377, 4125, 4160, 4165, 4170, 4171, 4185, 4190, 4196,  
4197, 4290, 4537, 4578, 5060, 5065, 5069, 5070, 6070, 6070
- \ctex\_fix\_varioref\_label:n ..... 3796, 3799, 3799
- \ctex\_fontset\_case:nnn .. 4488, 4488, 4887, 4952, 5387
- \ctex\_fontset\_case:nnnn .....  
..... 141, 142, 4494, 4494, 4748, 4812, 5075, 5319
- \ctex\_fontset\_error:n 4484, 4484, 4749, 4785, 4813,  
4850, 4917, 4981, 5077, 5081, 5082, 5119, 5320, 5352, 5419
- \ctex\_gadd\_hook:Nn ..... 6195, 6196, 6197, 6202
- \ctex\_gadd\_ltxhook:nn .....  
..... 426, 429, 2002, 2256, 6164, 6167, 6169, 6173, 6174
- \ctex\_gadd\_package\_hook:nnn .....  
..... 6147, 6160, 6170, 6195, 6195
- \ctex\_gadd\_selectfont\_hook:n .. 2251, 2255, 2259, 2263
- \ctex\_get\_macro\_meaning:N ..... 6339
- \ctex\_get\_macro\_meaning:NTF ..... 6335, 6361, 6417
- \ctex\_hang\_from:n ..... 3252, 3256, 3260
- \ctex\_heading\_depth:nn .. 3756, 3759, 3762, 3762, 3768
- \ctex\_heading\_glue:n ..... 3268, 3268, 3276
- \ctex\_heading\_hang:Nnn .. 3246, 3251, 3252, 3252, 3267
- \ctex\_hypersetup:n ..... 84,  
322, 2060, 2063, 2067, 2077, 2081, 2093, 2096, 2100, 2102
- \ctex\_if\_autoindent\_touched:TF .....  
..... 113, 113, 185, 190, 195, 4152
- \ctex\_if\_ccglue\_touched: ..... 2171
- \ctex\_if\_ccglue\_touched:TF ..... 2171, 2192, 2457
- \ctex\_if\_ccglue\_touched\_p: ..... 2171
- \ctex\_if\_format\_at\_least:nTF .....  
..... 48, 6068, 6068, 6125, 6162, 6172
- \ctex\_if\_platform\_macos:TF ..... 4505, 4510, 4510
- \ctex\_if\_preamble:TF ..... 97, 97, 98, 4549
- \ctex\_if\_rescanable:NnTF ..... 6359, 6359, 6393
- \ctex\_ignorespaces\_case:N .. 753, 753, 2325, 2328, 2331
- \ctex\_indent\_box:n ..... 2772, 2772, 2782
- \ctex\_load\_fontset: ..... 4512, 4512, 4544, 4557, 4577
- \ctex\_load\_std\_class:n ... 117, 118, 387, 390, 393, 396
- \ctex\_load\_zhmap:nnnn .....  
.. 516, 516, 523, 4779, 4844, 4911, 4975, 5112, 5346, 5413
- \ctex\_ltj\_add\_font\_features:n .. 1456, 1456, 1474, 1526
- \ctex\_ltj\_add\_font\_features:nn 1456, 1457, 1458, 1475
- \ctex\_ltj\_add\_kyenc:n ..... 990, 1259
- \ctex\_ltj\_char\_range\_key:nn ..... 1879, 1882, 1882
- \ctex\_ltj\_clear\_alt\_font:n ..... 1014, 1750
- \ctex\_ltj\_clear\_alternate\_font:n ..... 1796, 1802
- \ctex\_ltj\_declare\_alternate\_shape:nnnnnn .....  
..... 1711, 1731, 1731, 1733
- \ctex\_ltj\_declare\_char\_range:n .... 1834, 1837, 1839
- \ctex\_ltj\_declare\_char\_range:nn ... 1841, 1842, 1842
- \ctex\_ltj\_def\_char\_range\_key:n .... 1847, 1874, 1874
- \ctex\_ltj\_ensure\_default\_family: .. 1564, 1564, 2049
- \ctex\_ltj\_extract\_font: ..... 1075, 1082, 1082
- \ctex\_ltj\_family\_if\_exist:nN ..... 1368, 1381
- \ctex\_ltj\_family\_if\_exist:nNTF .....  
..... 1359, 1368, 1568, 1573, 1591, 1594, 1805
- \l\_ctex\_ltj\_family\_tl .....  
1361, 1366, 1423, 1424, 1439, 1443, 1446, 1457, 1799, 1800
- \ctex\_ltj\_fontspec:nn 1420, 1420, 1432, 1433, 1467, 1521
- \ctex\_ltj\_get\_and\_define\_fonts:nN .. 1167, 1167, 1175
- \ctex\_ltj\_get\_and\_define\_fonts\_al:nN ... 1171, 1174
- \ctex\_ltj\_get\_and\_define\_fonts\_ja:nN ... 1170, 1176
- \ctex\_ltj\_if\_alternate\_shape\_exist:n ..... 1118
- \ctex\_ltj\_if\_alternate\_shape\_exist:nTF .....  
..... 1085, 1100, 1118, 1127
- \ctex\_ltj\_if\_jfont:nTF .... 68, 1153, 1153, 1165, 1169
- \ctex\_ltj\_if\_jfont\_math:NTF ..... 1160, 1160, 1199
- \ctex\_ltj\_is\_kenc:n ..... 991, 1155
- \ctex\_ltj\_math\_group\_hook: ..... 67, 1198, 1204, 1224
- \ctex\_ltj\_patch\_external\_font:n ..... 992, 1124
- \ctex\_ltj\_pickup\_alt\_font:nn ..... 1018, 1129
- \ctex\_ltj\_pickup\_font: .....  
... 1050, 1052, 1070, 1070, 1081, 1143, 1179, 1181, 1183
- \ctex\_ltj\_reset\_alternate\_font:n ..... 1798, 1819
- \ctex\_ltj\_save\_alternate\_family:Nnnn .....  
..... 1676, 1697, 1700, 1700
- \ctex\_ltj\_save\_alternate\_seq:Nn 1647, 1652, 1652, 1683
- \ctex\_ltj\_save\_char\_range:n ..... 1845, 1852, 1852
- \ctex\_ltj\_save\_char\_range:nn ..... 1854, 1855
- \ctex\_ltj\_select\_alternate\_font: .. 1125, 1125, 2269
- \ctex\_ltj\_select\_font: ..... 1035, 1035, 2268
- \ctex\_ltj\_set\_alt\_font:nnnn ..... 1007, 1766, 1769
- \ctex\_ltj\_set\_alternate\_family:nn .. 1651, 1684, 1684
- \ctex\_ltj\_set\_alternate\_family:nnn .....  
..... 1696, 1704, 1706, 1706
- \ctex\_ltj\_set\_alternate\_family:Nnnnn 1688, 1693, 1699
- \ctex\_ltj\_set\_alternate\_family:nnTF 1289, 1637, 1637
- \ctex\_ltj\_set\_alternate\_prop:n .... 1616, 1621, 1621
- \ctex\_ltj\_set\_alternate\_shape:n 1757, 1764, 1771, 1776
- \ctex\_ltj\_set\_alternate\_shape:nnN .. 1743, 1772, 1772
- \ctex\_ltj\_set\_alternate\_shape:nnn .. 1746, 1752, 1752
- \ctex\_ltj\_set\_alternate\_shape:nnnn ..... 1754, 1755
- \ctex\_ltj\_set\_alternate\_shape:Nnnnnnn .....  
..... 1734, 1737, 1737
- \ctex\_ltj\_set\_family:nnn .....  
... 1281, 1281, 1451, 1498, 1515, 1533, 1540, 1546, 1552
- \ctex\_ltj\_set\_kanjiskip:N ..... 1912, 1912, 2165
- \ctex\_ltj\_set\_math\_letter:NN ..... 1206, 1218, 1226
- \ctex\_ltj\_set\_xkanjiskip:N .... 1912, 1914, 2282, 2366
- \ctex\_ltj\_subst\_font: ..... 1076, 1098, 1098
- \ctex\_ltj\_swap\_cs:NN ..... 1710, 1717, 1717
- \ctex\_ltj\_switch\_family:n .....  
..... 1356, 1357, 1367, 1424, 1452, 1512
- \ctex\_ltj\_update\_mathfont: ..... 1586, 1589, 1589
- \ctex\_ltj\_update\_mathfont:n ..... 1592, 1595, 1598
- \ctex\_ltj\_use\_jfont: ..... 1092, 1096, 1096
- \ctex\_ltj\_use\_math\_group:Nn ... 1192, 1192, 1205, 1223
- \ctex\_ltj\_zero\_globaldefs: ..... 893, 1030
- \ctex\_make\_spa:nn ..... 5828, 5828, 5836
- \ctex\_parse\_name:NN ..... 2242, 6265, 6269, 6269
- \ctex\_parse\_name:NNn ..... 6270, 6274, 6331

<code>\ctex_patch_boot:NnnTF</code> .....	<code>\ctex_set_upmap_unicode:nnn</code> .....
..... 6226, 6231, 6236, 6251, 6256, 6259, 6259	..... 1965, 1965, 1987, 5127, 5128, 5355, 5357, 5359
<code>\ctex_patch_cmd:Nnn</code> 563, 1933, 3507, 3520, 3526, 3532,	<code>\ctex_set_zhmap:n</code> ..... 421, 422, 445, 521, 1954, 1967
3605, 3608, 3613, 3619, 3630, 3637, 3640, 3643, 3646,	<code>\ctex_titleps_hook:</code> ..... 3516, 3518, 3551, 3552
3655, 3658, 3663, 3669, 3680, 4479, 4480, 4481, 6234, 6234	<code>\ctex_titlesec_hook:</code> ..... 3413, 3413, 3477
<code>\ctex_patch_cmd_all:NnnnTF</code> ..... 3828, 6229, 6229	<code>\ctex_undeclare_unicode_character:n</code> 525, 525, 529, 530
<code>\ctex_patch_cmd_once:NnnnTF</code> ... 1927, 2244, 6224, 6224	<code>\ctex_update_ccglue:</code> .... 2155, 2155, 2464, 2486, 2529
<code>\ctex_patch_failure:N</code> .....	<code>\ctex_update_ccwd:</code> ..... 2141, 2141, 2193, 2458, 2537
..... 942, 944, 1894, 2249, 3583, 3595, 3836, 6243, 6245	<code>\ctex_update_default_family:</code> 53, 578, 2030, 2031, 2031
<code>\ctex_plane_to_utfxvibe:Nn</code> ..... 540, 542, 559	<code>\ctex_update_em_unit:</code> ..... 2198, 2198, 2449, 2505
<code>\ctex_pop_file:</code> ..... 78, 537, 6070, 6074, 6085, 6099	<code>\ctex_update_kanjisize:</code> ..... 2296, 2318
<code>\ctex_preto_cmd:NnnTF</code> .... 934, 3580, 3591, 6249, 6249	<code>\ctex_update_parindent:</code> ..... 2427, 2488, 2488, 2530
<code>\ctex_provide_font_hook:NNN</code> .....	<code>\ctex_update_size:</code> ..... 2421, 2422, 2422, 2435
..... 1994, 1994, 2004, 2005, 2006, 2007	<code>\ctex_update_stretch:</code> ..... 2426, 2447, 2447
<code>\ctex_punct_map_bfseries:nn</code> .....	<code>\ctex_update_xkanjiskip:</code> ..... 2274, 2275, 2294
..... 786, 786, 4775, 4782, 4839,	<code>\ctex_update_ziju:</code> ..... 2429, 2503, 2503
4840, 4847, 4906, 4907, 4908, 4914, 4970, 4971, 4978,	<code>\ctex_use_package_hook:nn</code> .... 6203, 6203, 6214, 6220
5107, 5108, 5109, 5116, 5340, 5341, 5349, 5406, 5409, 5416	<code>\ctex_use_zhmap:</code> ..... 424, 430, 434, 440
<code>\ctex_punct_map_family:nn</code> ..... 781, 781, 4772,	<code>\ctex_varioref_hook:</code> ..... 3793, 3793, 3839, 3847
4773, 4774, 4781, 4836, 4837, 4838, 4846, 4902, 4903,	<code>\ctex_zhmap_case:nnn</code> ..... 142, 114,
4904, 4913, 4967, 4968, 4969, 4977, 5103, 5104, 5105,	114, 239, 244, 249, 4751, 4815, 4889, 4954, 5086, 5322, 5389
5114, 5115, 5336, 5337, 5338, 5348, 5405, 5408, 5410, 5415	<code>\ctex_zihao:n</code> ..... 4209, 4210, 4210
<code>\ctex_punct_map_itshape:nn</code> .....	<code>\ctex_ziju:n</code> ..... 2497, 2498
..... 803, 803, 4776, 4783, 4841, 4848,	<b>ctex internal commands:</b>
4905, 4915, 4972, 4979, 5106, 5117, 5339, 5350, 5407, 5417	<code>\_ctex_add_cmap_auxi:n</code> ..... 463, 465
<code>\ctex_punct_map_series:nnn</code> ..... 790, 791, 794	<code>\_ctex_add_cmap_auxii:Nn</code> ..... 466, 467
<code>\ctex_punct_set:n</code> . 763, 763, 4771, 4780, 4835, 4845,	<code>\g_ctex_after_end_preamble_hook_tl</code> 6182, 6186, 6194
4901, 4912, 4966, 4976, 5102, 5113, 5335, 5347, 5391, 5414	<code>\_ctex_assign_heading_name:nnn</code> ... 2732, 2734, 2735
<code>\ctex_punct_space:nn</code> ..... 812, 812, 814	<code>\l_ctex_autoindent_tl</code> .....
<code>\ctex_push_file:</code> ..... 71, 534, 6070, 6072, 6078, 6097	.. 112, 184, 189, 194, 2408, 2412, 2415, 2490, 2493, 2768
<code>\ctex_replace_package:nn</code> ..... 6125, 6127, 6135, 6139	<code>\_ctex_calc_bounds:nn</code> ..... 5885, 5886, 5890
<code>\ctex_save_font_size:nn</code> ..... 9, 4223, 4234	<code>\_ctex_ccglue:</code> ..... 2160, 2161, 2174, 2186
<code>\ctex_scheme_input:n</code> ..... 69, 69, 80, 3849, 4127	<code>\l_ctex_ccglue_skip</code> ..... 2159,
<code>\ctex_select_size:</code> .. 2409, 2416, 2419, 2420, 2441, 2501	2165, 2168, 2170, 2178, 2182, 2462, 2476, 2485, 2520, 2528
<code>\ctex_set:n</code> ..... 59, 65, 261, 2379, 2399, 4151, 4153	<code>\_ctex_char:N</code> ..... 592, 643
<code>\ctex_set:nn</code> .....	<code>\_ctex_char:NN</code> ..... 594, 644
.. 34, 59, 67, 139, 2389, 3852, 3910, 3938, 3970, 4005,	<code>\_ctex_char:NNN</code> ..... 608, 645
4022, 4036, 4052, 4060, 4068, 4076, 4083, 4084, 4085, 4585	<code>\_ctex_char:NNNN</code> ..... 625, 646
<code>\ctex_set_default_ccwd:Nn</code> .... 99, 99, 194, 2415, 2440	<code>\_ctex_char_aux:NN</code> ..... 597, 598, 600, 650, 654
<code>\ctex_set_deprecated_option:n</code> .....	<code>\_ctex_char_auxi:NNNN</code> ..... 611, 614
..... 134, 136, 173, 174, 228, 229, 230, 252, 257, 288, 289	<code>\_ctex_char_auxii:NNNNN</code> ..... 628, 631
<code>\ctex_set_font_size:Nnn</code> ..... 4364, 4364, 4379,	<code>\_ctex_char_auxiii:NNN</code> ..... 612, 615, 616, 657, 661
4387, 4398, 4409, 4410, 4411, 4412, 4413, 4414, 4415,	<code>\_ctex_char_auxiii:NNNN</code> ..... 629, 632, 633, 664, 668
4418, 4426, 4437, 4448, 4449, 4450, 4451, 4452, 4453, 4454	<code>\_ctex_char_preproc:w</code> ..... 673, 675
<code>\ctex_set_ignorespaces:</code> ..... 753, 755, 757, 759, 762	<code>\_ctex_char_raw:n</code> ..... 726, 749
<code>\ctex_set_jfm:n</code> .... 1231, 1231, 1236, 1247, 1251, 2348	<code>\_ctex_char_raw:nn</code> ..... 728, 746
<code>\ctex_set_upfamily:nnn</code> .....	<code>\_ctex_char_raw:nnn</code> ..... 733, 747
1938, 1938, 1985, 4795, 4796, 4797, 4798, 4860, 4861,	<code>\_ctex_char_raw:nnnn</code> ..... 739, 748
4862, 4863, 4927, 4928, 4929, 4930, 4931, 4932, 4991,	<code>\_ctex_char_raw_nn:n</code> ..... 746, 750
4992, 4993, 4994, 5129, 5130, 5131, 5132, 5133, 5134,	<code>\_ctex_char_raw_nnn:n</code> ..... 747, 751
5362, 5363, 5364, 5429, 5430, 5431, 5432, 5433, 5434, 5435	<code>\_ctex_char_raw_nnnn:n</code> ..... 748, 752
<code>\ctex_set_upfonts:nnnnnn</code> .....	<code>\_ctex_char_wrap:nn</code> ..... 710, 723
..... 1978, 1978, 1988, 4788, 4853, 4920, 4984, 5422	<code>\_ctex_char_wrap:nnn</code> ..... 712, 724
<code>\ctex_set_upmap:nnn</code> .....	<code>\_ctex_char_wrap:nnnn</code> ..... 714, 725
..... 1952, 1952, 1980, 1981, 1982, 1983,	<code>\c_ctex_class_tl</code> 72, 74, 120, 4114, 4120, 4122, 4123, 4125
1986, 4933, 4934, 5123, 5124, 5125, 5126, 5361, 5436, 5437	<code>\_ctex_clear_fntef_color:n</code> ... 2119, 2127, 2131, 2138

`\__ctex_cleveref_hook_aux:N` .....  
     ..... [3809](#), [3817](#), [3820](#), [3821](#), [3823](#), [3826](#)  
`\c__ctex_cmap_encoding_seq` ..... [448](#), [454](#)  
`\l__ctex_cmap_file_tl` ..... [475](#), [479](#), [489](#)  
`\l__ctex_ctexcap_options_clist` .... [6053](#), [6054](#), [6063](#)  
`\__ctex_def_heading_keys:n` ..... [2688](#), [2688](#), [2762](#)  
`\__ctex_default_ccwd_aux:n` ..... [100](#), [101](#)  
`\__ctex_default_ccwd_aux:w` ..... [104](#), [108](#)  
`\__ctex_disable_package_aux:n` ..... [6111](#), [6114](#)  
`\g__ctex_encoding_tl` ..... [110](#), [218](#), [221](#), [224](#), [373](#),  
     [377](#), [507](#), [512](#), [581](#), [587](#), [647](#), [2075](#), [2547](#), [4155](#), [4181](#), [4195](#)  
`\g__ctex_end_preamble_hook_tl` ..... [6180](#), [6184](#), [6193](#)  
`\c__ctex_engine_file_str` ..... [23](#), [2377](#)  
`\c__ctex_engine_str` ..... [23](#)  
`\c__ctex_everysel_loaded_bool` ..... [46](#), [1989](#), [2208](#)  
`\g__ctex_expl_status_seq` ..... [6080](#), [6088](#), [6094](#)  
`\l__ctex_expl_status_tl` ..... [6088](#), [6089](#), [6093](#)  
`\l__ctex_family_default_init_tl` ..... [2033](#), [2051](#)  
`\__ctex_family_default_wrap:n` ..... [2036](#), [2052](#), [2055](#)  
`\g__ctex_font_size_int` ..... [151](#),  
     [167](#), [168](#), [169](#), [340](#), [344](#), [381](#), [4172](#), [4173](#), [4198](#), [4199](#), [4372](#)  
`\c__ctex_font_size_prop` ..... [4212](#), [4223](#), [4339](#), [4366](#)  
`\c__ctex_font_size_seq` ..... [4221](#), [4320](#)  
`\g__ctex_font_size_system_tl` .....  
     ..... [160](#), [4238](#), [4288](#), [4291](#), [4295](#)  
`\l__ctex_font_size_temp_prop` .. [4223](#), [4227](#), [4236](#), [4319](#)  
`\l__ctex_font_size_temp_seq` ... [4224](#), [4232](#), [4237](#), [4321](#)  
`\l__ctex_font_size_tl` [4212](#), [4213](#), [4331](#), [4332](#), [4366](#), [4367](#)  
`\g__ctex_fontset_tl` [227](#), [4503](#), [4506](#), [4507](#), [4514](#), [4518](#),  
     [4519](#), [4522](#), [4523](#), [4526](#), [4532](#), [4537](#), [4554](#), [4556](#), [4561](#), [4576](#)  
`\__ctex_get_font_sizes:Nn` ..... [4331](#), [4334](#), [4334](#)  
`\__ctex_get_macro_meaning:w` ..... [6335](#), [6341](#), [6344](#)  
`\l__ctex_heading_bool` ..... [265](#), [4110](#)  
`\c__ctex_heading_level_prop` ..... [3764](#), [3769](#)  
`\l__ctex_heading_skip` .... [2789](#), [2796](#), [2798](#), [2800](#),  
     [2801](#), [2816](#), [2817](#), [2820](#), [2822](#), [2823](#), [2826](#), [3271](#), [3272](#), [3273](#)  
`\c__ctex_headings_cs_seq` ..... [3360](#), [3374](#), [3384](#), [3407](#)  
`\c__ctex_headings_seq` ... [2649](#), [2759](#), [3490](#), [3564](#), [3795](#)  
`\__ctex_hookto_cmd:Nnnnw` ..... [6251](#), [6256](#), [6414](#), [6414](#)  
`\__ctex_hookto_cmd_parameter:Nnnnw` . [6421](#), [6441](#), [6441](#)  
`\__ctex_hookto_cmd_parameterless:Nnnnw` .....  
     ..... [6420](#), [6426](#), [6426](#)  
`\__ctex_if_font_exists:n` ..... [5857](#)  
`\__ctex_if_font_exists:nTF` ..... [5840](#), [5857](#)  
`\l__ctex_if_otf_exists_tl` [5859](#), [5864](#), [5870](#), [5875](#), [5879](#)  
`\l__ctex_if_ttf_exists_tl` [5860](#), [5866](#), [5872](#), [5876](#), [5879](#)  
`\__ctex_initial_heading:n` ..... [2658](#), [2658](#), [2761](#)  
`\__ctex_insert_indent:` ..... [2775](#), [2777](#), [3258](#)  
`\__ctex_left_hook_aux:nn` ..... [6439](#)  
`\l__ctex_line_spread_fp` .....  
     ..... [129](#), [177](#), [4174](#), [4175](#), [4200](#), [4201](#), [4458](#), [4460](#)  
`\l__ctex_line_stretch_tl` ..... [2440](#), [2445](#), [2446](#), [2450](#)  
`\l__ctex_ltj_add_alternate_bool` [1436](#), [1448](#), [1466](#), [1473](#)  
`\__ctex_ltj_alternate_cs:n` .....  
     .... [1316](#), [1319](#), [1320](#), [1321](#), [1351](#), [1352](#), [1354](#), [1439](#),  
     [1442](#), [1443](#), [1445](#), [1446](#), [1647](#), [1689](#), [1749](#), [1768](#), [1808](#), [1825](#)  
`\l__ctex_ltj_alternate_family_tl` ..... [1713](#), [1716](#)  
`\l__ctex_ltj_alternate_options_clist` .....  
     ..... [1666](#), [1667](#), [1678](#), [1682](#)  
`\l__ctex_ltj_alternate_prop` .....  
     ..... [1285](#), [1632](#), [1633](#), [1636](#), [1646](#), [1654](#)  
`\l__ctex_ltj_base_CJKfamily_tl` .....  
     ..... [1286](#), [1300](#), [1349](#), [1749](#), [1768](#)  
`\l__ctex_ltj_base_family_tl` ..... [78](#),  
     [1317](#), [1344](#), [1350](#), [1695](#), [1735](#), [1788](#), [1805](#), [1811](#), [1812](#), [1824](#)  
`\l__ctex_ltj_char_range_clist` .....  
     ... [1284](#), [1618](#), [1639](#), [1665](#), [1677](#), [1690](#), [1712](#), [1734](#), [1885](#)  
`\g__ctex_ltj_char_range_prop` ..... [1741](#), [1842](#)  
`\l__ctex_ltj_char_range_tl` .....  
     ..... [1741](#), [1744](#), [1844](#), [1846](#), [1848](#), [1850](#), [1857](#)  
`\__ctex_ltj_chardef_text_cmd:` ..... [902](#), [906](#)  
`\__ctex_ltj_chardef_text_cmd_aux:N` ..... [907](#), [908](#)  
`\__ctex_ltj_check_family:n` ..... [1311](#), [1311](#), [1645](#)  
`\l__ctex_ltj_cmd_tl` ..... [901](#), [905](#), [907](#)  
`\l__ctex_ltj_current_font_tl` .....  
     ..... [64](#), [1038](#), [1042](#), [1043](#), [1056](#), [1104](#), [1109](#)  
`\l__ctex_ltj_current_shape_tl` . [1127](#), [1130](#), [1133](#), [1134](#)  
`\__ctex_ltj_declare_char_range:nn` .... [1838](#), [1840](#)  
`\g__ctex_ltj_default_features_clist` [1304](#), [1556](#), [1557](#)  
`\__ctex_ltj_family_csname:n` ... [1315](#), [1332](#), [1346](#), [1373](#)  
`\g__ctex_ltj_family_font_name_prop` .....  
     ..... [1291](#), [1308](#), [1313](#), [1384](#), [1460](#), [1566](#), [1578](#)  
`\g__ctex_ltj_family_font_options_prop` .....  
     ..... [1292](#), [1308](#), [1463](#)  
`\g__ctex_ltj_family_int` . [1426](#), [1428](#), [1727](#), [1728](#), [1730](#)  
`\g__ctex_ltj_family_name_prop` .....  
     ..... [1308](#), [1317](#), [1338](#), [1370](#), [1695](#)  
`\__ctex_ltj_family_unknown_warning:n` .....  
     ..... [1364](#), [1382](#), [1382](#), [1816](#)  
`\l__ctex_ltj_font_options_clist` .... [1288](#), [1293](#),  
     [1294](#), [1295](#), [1301](#), [1337](#), [1464](#), [1465](#), [1468](#), [1686](#), [1687](#), [1691](#)  
`\g__ctex_ltj_fontname_seq` ..... [1061](#), [1066](#), [1069](#)  
`\__ctex_ltj_fontspec:nnn` ..... [1427](#), [1434](#), [1454](#)  
`\g__ctex_ltj_fontspec_family_tl` .....  
     ..... [1336](#), [1339](#), [1345](#), [1350](#), [1375](#)  
`\g__ctex_ltj_fontspec_prop` ..... [1422](#), [1450](#), [1455](#)  
`\__ctex_ltj_gset_family_cs:nn` ..... [1296](#), [1330](#), [1330](#)  
`\__ctex_ltj_if_alt_set:nTF` ..... [1029](#), [1120](#)  
`\__ctex_ltj_if_jfont_math:w` ..... [1161](#), [1164](#)  
`\l__ctex_ltj_jfm_tl` ..... [1231](#), [1262](#), [1264](#), [1306](#)  
`\__ctex_ltj_ltj_if_jfont:w` ..... [1155](#), [1158](#)  
`\c__ctex_ltj_math_fam_int` ..... [1270](#), [1271](#)  
`\c__ctex_ltj_math_family_tl` .....  
     ..... [1600](#), [1602](#), [1604](#), [1607](#), [1611](#)  
`\c__ctex_ltj_math_tl` .....  
     ... [1265](#), [1266](#), [1268](#), [1270](#), [1552](#), [1591](#), [1601](#), [1606](#), [1610](#)  
`\__ctex_ltj_msg_def_family_map:n` ..... [1397](#), [1399](#)  
`\__ctex_ltj_msg_family_map:n` ..... [1329](#), [1396](#), [1410](#)  
`\__ctex_ltj_noembed_wrap:n` ..... [1278](#), [1280](#)  
`\__ctex_ltj_pass_args:nnnn` .....  
     ... [1481](#), [1481](#), [1497](#), [1514](#), [1520](#), [1532](#), [1539](#), [1545](#), [1551](#)  
`\__ctex_ltj_patch_external_font:w` . [1088](#), [1123](#), [1123](#)

<code>\__ctex_ltj_patch_text_cmd:NN</code> .....	928, 931	<code>\__ctex_plus_key_aux:nn</code> .....	
<code>\__ctex_ltj_pickup_alt_font:Nn</code> .....	1024, 1151	.....	2714, 2716, 2717, 2718, 2719, 2720, 2721, 2724
<code>\__ctex_ltj_pop_fontname:</code> .....	1055, 1064, 1145	<code>\l__ctex_prefix_str</code> .....	6332, 6350, 6371, 6428, 6429
<code>\__ctex_ltj_post_arg:w</code> .....	1484, 1490	<code>\l__ctex_prefix_tl</code> .....	6371, 6376, 6387
<code>\__ctex_ltj_push_alternate_prop:n</code> .....	1622, 1623	<code>\__ctex_provide_font_hook_aux:NNNN</code> .....	1996, 1999
<code>\__ctex_ltj_push_alternate_prop:nn</code> .....	1627, 1630, 1886	<code>\l__ctex_punct_bounds_clist</code> ...	5844, 5853, 5883, 5889
<code>\__ctex_ltj_push_fontname:n</code> .....	1051, 1059, 1142	<code>\c__ctex_punct_family_clist</code> .....	765, 775
<code>\c__ctex_ltj_range_max_int</code> .....	1869, 1873	<code>\l__ctex_punct_font</code> .....	5841, 5842, 5843
<code>\c__ctex_ltj_range_min_int</code> .....	1868, 1872	<code>\c__ctex_punct_seq</code> .....	5820, 5845
<code>\__ctex_ltj_range_normalization:nn</code> .....	1759, 1858, 1860	<code>\l__ctex_punct_tl</code> .....	
<code>\l__ctex_ltj_redirect_jfm_prop</code> .....	1231	.....	254, 574, 575, 822, 1251, 2345, 2346, 2347, 2348
<code>\g__ctex_ltj_reset_alternate_prop</code> .....		<code>\__ctex_rebuild_cmd:w</code> ...	6363, 6364, 6368, 6443, 6444
.....	1322, 1810, 1822, 1830	<code>\l__ctex_replacement_str</code> .....	
<code>\__ctex_ltj_save_alternate_auxi:w</code> .....	1655, 1657	.....	6332, 6352, 6373, 6411, 6412, 6446
<code>\__ctex_ltj_save_alternate_auxii:w</code> .....	1660, 1661, 1663	<code>\l__ctex_replacement_tl</code> .....	6373, 6378, 6389
<code>\__ctex_ltj_save_alternate_shape:Nn</code> .....		<code>\__ctex_restore_selectfont:</code> .....	2217, 2239, 2248
.....	1748, 1767, 1784, 1784, 1792	<code>\__ctex_right_hook_aux:nn</code> .....	6440
<code>\__ctex_ltj_select_font_aux:</code> ..	1039, 1045, 1045, 1057	<code>\__ctex_save_bounds:n</code> .....	5847, 5881
<code>\__ctex_ltj_set_alternate_family:n</code> .....	1340, 1347	<code>\__ctex_save_cmap:Nn</code> .....	469, 472
<code>\__ctex_ltj_set_alternate_family_aux:nnn</code> .....	1641, 1650	<code>\__ctex_save_font_size:nn</code> .....	
<code>\__ctex_ltj_set_family_aux:nnn</code> .....	1640, 1643	.....	4223, 4225, 4234, 4261, 4284, 4315
<code>\__ctex_ltj_tmp:w</code> .....	1719, 1721, 1722	<code>\l__ctex_scheme_tl</code> .....	276, 3849, 4127
<code>\l__ctex_ltj_tmp_clist</code> .....	1625, 1626, 1628, 1635	<code>\g__ctex_section_depth_int</code> .....	
<code>\l__ctex_ltj_tmp_tl</code> .....	1066, 1067, 1287,	.....	81, 270, 272, 4050, 4066, 4082
1288, 1313, 1324, 1327, 1461, 1469, 1505, 1512, 1515,		<code>\c__ctex_section_headings_seq</code> .....	2645, 3315, 3431
1568, 1573, 1591, 1592, 1594, 1595, 1669, 1671, 1672, 1679		<code>\__ctex_set_font_size:nnNn</code> .....	4367, 4370
<code>\g__ctex_ltj_unknown_family_seq</code> ...	1386, 1388, 1393	<code>\g__ctex_spa_iow</code> .....	5830, 5833, 5835, 5850
<code>\__ctex_ltj_update_family_uid:N</code> .....	1294, 1686, 1724, 1725	<code>\c__ctex_std_class_tl</code> .....	124, 4098, 4113, 4114, 4179, 4202
<code>\__ctex_ltj_use_global_options:N</code> .....		<code>\g__ctex_std_options_clist</code> .....	117, 345, 357, 383, 384
.....	1295, 1302, 1302, 1667, 1687	<code>\__ctex_titlesec_format:n</code> .....	3427, 3435, 3438
<code>\g__ctex_mac_baoli_dir_tl</code> .....	5300, 5303	<code>\__ctex_titlesec_format:Nn</code> .....	3417, 3441, 3443
<code>\g__ctex_mac_kaiti_dir_tl</code> .....	5249, 5259, 5261, 5290, 5294	<code>\__ctex_titlesec_format_explicit:Nn</code> .....	3418, 3453
<code>\g__ctex_mac_pingfang_bool</code> .....	5150, 5174, 5280, 5471	<code>\__ctex_titlesec_hook:n</code> .....	3427, 3431, 3433
<code>\g__ctex_mac_pingfang_dir_tl</code> .....	5268, 5273, 5279	<code>\__ctex_titlesec_spacing:Nn</code> .....	3436, 3463
<code>\g__ctex_mac_stfangso_dir_tl</code> .....	5282, 5286, 5288	<code>\__ctex_titlesec_spacing:nnnnn</code> .....	3464, 3465
<code>\g__ctex_mac_yuanti_dir_tl</code> .....	5305, 5309	<code>\__ctex_tmp:N</code> .....	2213, 2242
<code>\g__ctex_macos_ver_found_bool</code> .....	5016, 5023, 5033	<code>\__ctex_tmp:w</code> .....	6272, 6325, 6336, 6357
<code>\g__ctex_macos_ver_ior</code> .....	5015, 5019, 5021, 5036	<code>\l__ctex_tmp_box</code> .....	
<code>\g__ctex_macos_ver_tl</code> .....	5010, 5028, 5062, 5068	.....	55, 2144, 2145, 3263, 3264, 3265, 3568, 3572
<code>\c__ctex_msyh_suffix_tl</code> .....		<code>\l__ctex_tmp_dim</code> .....	55,
.....	5379, 5394, 5395, 5401, 5402, 5426, 5427	2450, 2451, 2460, 2463, 2472, 2510, 2512, 2514, 2523, 2525	
<code>\c__ctex_package_cctab</code> .....	6083, 6098, 6102	<code>\l__ctex_tmp_int</code> .....	55, 544,
<code>\__ctex_package_loaded_warning:nn</code> .....	6146, 6149	546, 547, 549, 553, 555, 2471, 2473, 2474, 2480, 2481,	
<code>\l__ctex_parameter_str</code> .....	6332, 6351, 6372, 6419	2512, 2513, 2514, 2517, 2518, 2523, 2525, 3314, 3317, 3321	
<code>\l__ctex_parameter_tl</code> .....	6372, 6377, 6388	<code>\l__ctex_tmp_tl</code> .....	55, 329, 336, 351, 352,
<code>\__ctex_parse_name:nnNNnN</code> .....	6284, 6293	2215, 2244, 2690, 2744, 2757, 2764, 3764, 3765, 4339, 4340	
<code>\__ctex_parse_name:w</code> .....	6306, 6312, 6321	<code>\__ctex_update_stretch_auxi:</code> .....	2452, 2455
<code>\l__ctex_patch_cleveref_bool</code> .....	3806, 3811	<code>\__ctex_update_stretch_auxii:</code> .....	2190, 2453, 2467
<code>\__ctex_patch_cmd:Nnnnw</code> .....	6226, 6231, 6236, 6390, 6390	<code>\__ctex_update_stretch_auxiii:</code> ...	2194, 2468, 2469
<code>\__ctex_patch_false:w</code> .....	6262, 6268, 6405, 6407, 6424, 6450	<code>\__ctex_use_package_hook_aux:N</code> .....	6206, 6208
<code>\__ctex_patch_rebuild:Nn</code> .....		<code>\__ctex_write_family:nn</code> .....	153, 5832, 5837
.....	6363, 6369, 6369, 6402, 6443, 6447	<code>\l__ctex_xkanjiskip_skip</code> .....	2274, 2365, 2366, 2367
<code>\__ctex_patch_replace:nnTF</code> .....	6397, 6409, 6409	<code>\l__ctex_xkanjiskip_tl</code> .....	2281, 2286, 2287, 2364
<code>\__ctex_patch_rescan:NNn</code> .....	6371, 6372, 6373, 6381	<code>\c__ctex_zero_tl</code> .....	2424, 2431, 2433
<code>\__ctex_patch_toc_width:n</code> .....	3575, 3584, 3600, 3601	<code>\g__ctex_zhmap_tl</code> .....	423, 425, 444
<code>\__ctex_patch_true:w</code> .....	6261, 6267, 6403, 6436, 6448	<code>\g__ctex_zhmCJK_bool</code> .....	111, 238, 243, 248, 504, 579, 1993

`\l__ctex_ziju_dim` 2506, 2507, 2511, 2522, 2524, 2528, 2532  
`\l__ctex_ziju_tl` ..... 2424, 2432, 2433, 2500, 2506  
`ctex.newluacmd` ..... 969  
`\ctexbibname` ..... 2618, 2638  
`\CTEXcounter` ..... 2553  
`\CTEXdigits` ..... 31, 2563  
`CTEXfilltwosides` ..... 36, 38  
`\CTEXifname` ..... 29, 2808, 2898, 2940, 3082, 3294, 3302, 3310, 3559, 3607, 3610, 3615, 3627, 3632, 3639, 3642, 3645, 3648, 3657, 3660, 3665, 3677, 3682  
`\CTEXindent` ..... 36, 2533  
`\CTEXnoindent` ..... 36, 2533  
`\CTEXnumber` ..... 31, 2563  
`\CTEXnumberline` ..... 3308, 3328, 3928  
`\CTEXoptions` ..... 37, 2381  
**ctexpar internal commands:**  
`\__ctexpar` ..... 3204  
`\ctexset` ..... 5, 2378, 2388, 2398  
`\CTEXsetfont` ..... 36, 2419  
`\CTEXsetup` ..... 37, 2381  
`\CTEXsout` ..... 36, 38  
`\ctexspadef` ..... 812, 5852  
`\CTEXthechapter` ..... 28, 3025, 3303, 3483, 3627, 3645, 3677  
`\CTEXtheparagraph` ..... 28  
`\CTEXthepart` ..... 28, 3295, 3488  
`\CTEXthesection` ..... 28, 3607, 3615, 3632, 3639, 3648, 3657, 3665, 3682  
`\CTEXthesubparagraph` ..... 28  
`\CTEXthesubsection` ..... 28, 3610, 3642, 3660  
`\CTEXthesubsubsection` ..... 28  
`\CTEXunderdblline` ..... 36, 38  
`\CTEXunderdot` ..... 36, 38  
`\CTEXunderline` ..... 36, 38  
`\CTEXunderwave` ..... 36, 38  
`\CTEXxout` ..... 36, 38  
`\CurrentOption` ..... 357  
`\cwd` ..... 2296

**D**

`\d` ..... 5027  
`declarecharrange` ..... 32, 1832  
`\DeclareErrorKanjiFont` ..... 1921  
`\DeclareFontEncoding` ..... 1253  
`\DeclareFontFamily` ..... 490, 1260, 4596, 4597, 4598, 4599, 4600, 4601  
`\DeclareFontShape` 1261, 1263, 1942, 1943, 1946, 1947, 1948, 1949, 4610, 4611, 4612, 4613, 4614, 4615, 4616, 4617, 4618, 4621, 4622, 4623, 4624, 4625, 4626, 4627, 4628, 4629, 4632, 4633, 4634, 4635, 4636, 4637, 4638, 4639, 4640, 4643, 4644, 4645, 4646, 4647, 4648, 4649, 4650, 4651, 4656, 4657, 4658, 4659, 4660, 4661, 4662, 4663, 4664, 4667, 4668, 4669, 4670, 4671, 4672, 4673, 4674, 4675, 4678, 4679, 4680, 4681, 4682, 4683, 4684, 4685, 4686, 4689, 4690, 4691, 4692, 4693, 4694, 4695, 4696, 4697, 4702, 4703, 4704, 4705, 4706, 4707, 4708, 4709, 4710, 4713, 4714, 4715, 4716, 4717, 4718, 4719, 4720,

4721, 4724, 4725, 4726, 4727, 4728, 4729, 4730, 4731, 4732, 4735, 4736, 4737, 4738, 4739, 4740, 4741, 4742, 4743  
`\DeclareFontSubstitution` ..... 1256  
`\DeclareKanjiFamily` ..... 1940, 1941, 4602, 4603, 4604, 4605, 4606, 4607  
`\DeclareKanjiSubstitution` ..... 1922, 1923  
`\DeclareMathSizes` ..... 4332  
`\DeclareSymbolFont` ..... 1266, 1601, 1924  
`\def` ..... 407, 1900, 2871, 2912, 2948, 2969, 2985, 3016, 3057, 3067, 3092, 3112, 3138, 3142, 3179, 3194, 3210, 3211, 3215, 4392, 4403, 4431, 4442, 5492, 5528  
`\defaultCJKfontfeatures` ..... 1530  
`\defbeamertemplate` ..... 3691, 3710, 3729  
`\defbeamertemplatealias` ..... 3748, 3749, 3750  
`\detokenize` ..... 408  
**dim commands:**  
`\dim_abs:n` ..... 2524  
`\dim_add:Nn` ..... 2507  
`\dim_compare:nNnTF` ..... 2451, 2492, 2508, 2515, 2779, 2788, 3272  
`\dim_eval:n` ..... 2478, 2523  
`\dim_max:nn` ..... 3571  
`\dim_min:nn` ..... 2524  
`\dim_new:N` ..... 58, 2154, 2297, 2298, 2299, 2532  
`\dim_set:Nn` ..... 2145, 2199, 2200, 2201, 2450, 2460, 2493, 2506, 2510, 2538, 2774, 2795, 3254, 3569, 4462  
`\dim_sub:Nn` ..... 2514  
`\dim_to_decimal:n` ..... 4229, 4230, 4341  
`\dim_to_decimal_in_unit:nn` ..... 5896  
`\dim_use:N` ..... 105, 4462  
`\dim_zero:N` ..... 2544  
`\c_max_dim` ..... 2451, 2788  
`\c_zero_dim` ..... 2463, 2478, 2492, 2508, 2515, 2703, 2779, 3272, 4083  
`\directlua` ..... 871, 875  
`\do` ..... 412, 1904  
`\document` ..... 6187, 6190  
`\dospecials` ..... 1904

**E**

`\edef` ..... 5511, 5512  
`\else` ..... 402, 405, 406, 410, 1899, 1902, 1929, 1930, 1934, 1935, 2843, 2852, 2861, 2877, 2882, 2918, 2923, 3004, 3012, 3028, 3034, 3040, 3053, 3063, 3123, 3128, 3147, 3152, 3173, 3178, 3187, 3209, 3225, 3238, 3891, 3919, 5523, 5524, 5552, 5576, 5625, 5674, 5698, 5730, 5779  
**else commands:**  
`\else:` ..... 2175  
`\em` ..... 1927  
`\emminershape` ..... 1929, 1930, 1935, 1937  
`\end` ..... 3707, 3726, 3745, 5965  
`\endcsname` ..... 401, 404, 405, 3130, 3139, 3140, 3141, 3148, 3168, 3169, 3171, 3174, 3183, 3184, 3185, 3188, 3833, 3834, 5518, 5520, 5521, 5522, 5525, 5527, 5531  
`\endgroup` ..... 402, 419, 2903, 2944, 2960, 2981, 3086, 3105, 3170, 3208, 3233, 3709, 3728, 3747, 5511, 5519, 5534

\endinput ..... 402, 5516  
 \endlinechar ..... 5490, 5491, 5869, 5871  
 \everypar ..... 3127, 3228, 3240  
 \EverySelectfont ..... 2252  
**exp commands:**  
 \exp\_after:wN ..... 104,  
 907, 1072, 1088, 1090, 1161, 3464, 4213, 4332, 4367, 6341  
 \exp\_args:cc ..... 3789  
 \exp\_args:Nc ..... 466, 1996, 3436, 3441, 6295  
 \exp\_args:Nco ..... 3557  
 \exp\_args:Ne ..... 463, 474, 506,  
 545, 584, 593, 677, 819, 2002, 2497, 2546, 2804, 4209, 4460  
 \exp\_args:NNc ..... 928, 1037, 6205  
 \exp\_args:Nnc ..... 117  
 \exp\_args:NNe ..... 2690, 4318, 4320  
 \exp\_args:Nne ..... 649, 656, 663, 3802, 6118  
 \exp\_args:NNNo ..... 4461  
 \exp\_args:NNo ..... 2764  
 \exp\_args:Nno ..... 492  
 \exp\_args:No . 121, 351, 674, 1169, 1627, 3460, 5847, 6357  
 \exp\_last\_unbraced:Ne ..... 602, 618, 635, 6306  
 \exp\_last\_unbraced:NNNo ..... 107  
 \exp\_last\_unbraced:NNo ..... 2243  
 \exp\_not:N . 575, 586, 587, 588, 652, 653, 659, 660, 666,  
 667, 922, 923, 1062, 1109, 1141, 1256, 1338, 1339, 1676,  
 2041, 2042, 2043, 2055, 2664, 2665, 2666, 2671, 2673,  
 2674, 2675, 2676, 2679, 2680, 2683, 2695, 2696, 2697,  
 2698, 2699, 2708, 2709, 2710, 2712, 2713, 2726, 2728,  
 2730, 3320, 3322, 3323, 3324, 3325, 3448, 3449, 3458,  
 3459, 3496, 3498, 3499, 3802, 4532, 6286, 6287, 6288, 6376  
 \exp\_not:n ..... 103, 144, 654, 661,  
 668, 1335, 1336, 1337, 1468, 1469, 1677, 1678, 1679,  
 1704, 1760, 1761, 1789, 2036, 2056, 2205, 3467, 3790,  
 6055, 6190, 6261, 6262, 6304, 6376, 6377, 6378, 6433, 6434  
 \expandafter ..... 401, 402, 404, 405, 413, 417, 3834,  
 5493, 5494, 5518, 5519, 5520, 5521, 5522, 5527, 5531, 5532  
 experiment ..... 2353  
 experiment/CJKecglue ..... 13, 2353  
 experiment/font-size-system ..... 9, 159  
 \ExplSyntaxOff  
 .. 849, 935, 939, 1928, 2245, 3580, 3690, 3830, 5902, 6239  
 \ExplSyntaxOn ..... 886, 3751, 5813

**F**

\familydefault ..... 2039  
 fancyhdr ..... 36, 311  
 \fangsong ..... 5452  
 \fi ..... 402, 405,  
 406, 414, 415, 416, 418, 1902, 1909, 2834, 2839, 2845,  
 2854, 2863, 2866, 2881, 2886, 2896, 2907, 2922, 2927,  
 2938, 2964, 2988, 2996, 2997, 3000, 3004, 3014, 3032,  
 3038, 3044, 3056, 3066, 3070, 3079, 3080, 3090, 3095,  
 3109, 3113, 3119, 3125, 3132, 3134, 3156, 3157, 3175,  
 3189, 3192, 3213, 3222, 3241, 3242, 3606, 3609, 3614,  
 3624, 3625, 3631, 3638, 3641, 3644, 3647, 3656, 3659,

3664, 3674, 3675, 3681, 3703, 3722, 3741, 3891, 3919,  
 5523, 5524, 5533, 5571, 5595, 5652, 5693, 5725, 5749, 5806  
**fi commands:**  
 \fi: ..... 2028, 2175  
 \figurename ..... 2599  
 figurename ..... 14  
**file commands:**  
 \g\_file\_curr\_name\_str ..... 834  
 \file\_get\_full\_name:nTF ..... 474  
 \file\_if\_exist:nTF .....  
 ..... 28, 4288, 4511, 4526, 5017, 5059, 5380, 5383  
 \file\_if\_exist\_input:nTF ..... 74  
 \file\_if\_exist\_p:n ..... 153, 5863, 5865  
 \file\_input:n ..... 157, 75, 77, 536, 4374, 4375, 6073  
 \file\_input\_stop: ..... 6176  
 \fmtversion ..... 6069  
 fntef ..... 36, 38, 292  
 \font ..... 4596, 4597, 4598, 4599, 4600, 4601  
 fontset ..... 7, 227, 4545  
 \fontsize ..... 4213  
**fontspec commands:**  
 \fontspec\_font\_if\_exist:nTF .....  
 ..... 5153, 5170, 5177, 5183, 5188, 5190  
 \fontspec\_gset\_family:Nnn ..... 1336  
 \fontspec\_set\_family:Nnn ..... 1713  
**fontspec internal commands:**  
 \\_fontspec\_fontname\_wrap:n ..... 69, 1278  
 \footnotesep ..... 4461  
 \footnotesize ..... 4398, 4437, 4461  
**fp commands:**  
 \fp\_eval:n ..... 2500, 5892  
 \fp\_if\_nan:nTF ..... 4174, 4200, 4458  
 \fp\_set:Nn ..... 4175, 4201  
 \fp\_use:N ..... 2431, 4460  
 \c\_nan\_fp ..... 178  
 \c\_zero\_fp ..... 2431

**G**

GBK ..... 6, 38, 215  
 \gdef . 409, 415, 3346, 3347, 3348, 3349, 3352, 3353, 3354, 3355  
 \global ..... 3008, 3227, 3230  
 \globaljfont ..... 1090  
 \glueexpr ..... 3697, 3708, 3716, 3727, 3735, 3746  
**group commands:**  
 \group\_begin: . 671, 892, 965, 1037, 1047, 1074, 1102,  
 1140, 1162, 1208, 1283, 1334, 1708, 1786, 1804, 1821,  
 2035, 2212, 2300, 2670, 2678, 2742, 3270, 3370, 3516,  
 3575, 4235, 4461, 5839, 6184, 6205, 6263, 6271, 6280, 6335  
 \group\_end: 676, 897, 1034, 1037, 1053, 1078, 1115, 1144,  
 1166, 1212, 1298, 1341, 1714, 1790, 1817, 1828, 2047,  
 2243, 2317, 2681, 2685, 2764, 3274, 3387, 3539, 3579,  
 4322, 4461, 5855, 6184, 6205, 6283, 6330, 6358, 6392, 6416  
 \group\_insert\_after:N ..... 1112, 1113  
 \gtfamily ..... 1929, 1934

**H**

\hbox ..... 1902

hbox commands:  
 \hbox\_set:Nn ..... 2144, 3263, 3568  
 heading ..... 10, 265  
 \heiti ..... 5452  
 hook commands:  
 \hook\_gput\_code:nnn ..... 6165  
 \hskip ..... 3165, 3181, 3203, 3210, 3236, 3638, 3641  
 \hspace ..... 2876, 2917, 3295, 3303, 3488  
 \Huge ..... 2941, 2978, 3083, 3102, 3895, 3926, 4415, 4454  
 \huge ..... 2899,  
 2937, 2957, 3078, 3877, 3894, 3898, 3924, 3932, 4414, 4453  
 hyperref ..... 36, 316  
 \hypersetup ..... 321, 322, 2064  
 \hyphenchar ..... 4596, 4597, 4598, 4599, 4600, 4601

I

\if ..... 408  
 if commands:  
 \if\_bool:N ..... 1993  
 \if\_false: ..... 96  
 \if\_meaning:w ..... 2174  
 \if\_true: ..... 95  
 \ifctexpdf ..... 94  
 \ifdim ..... 3117, 3160, 3198, 3217  
 \ifmmode ..... 1902  
 \ifnum ..... 406, 2872, 2894, 2913, 2936, 3017, 3076, 3143,  
 3173, 3187, 3606, 3609, 3614, 3621, 3622, 3631, 3638,  
 3641, 3644, 3647, 3656, 3659, 3664, 3671, 3672, 3681, 5523  
 \ifodd ..... 403, 415, 2839, 2841, 2866, 2873, 2907,  
 2914, 2964, 2988, 3010, 3021, 3070, 3090, 3095, 3109,  
 3121, 3132, 3148, 3161, 3199, 3218, 3222, 3701, 3720, 3739  
 \ifx ..... 401, 404, 405, 5521, 5522, 5527  
 \ifzhmappdf .. 5519, 5550, 5574, 5598, 5655, 5696, 5728, 5752  
 \ignorespaces ..... 3243  
 \immediate ..... 5530  
 indent ..... 35, 197  
 \indexname ..... 2602  
 indexname ..... 14  
 \input ..... 5812, 5905  
 \insertcontinuationtext ..... 2611, 2620, 2627  
 \insertpart ..... 3705, 3706  
 \insertromanpartnumber ..... 3695, 3864  
 \insertsection ..... 3724, 3725  
 \insertsectionnumber ..... 3714, 3962  
 \insertsubsection ..... 3743, 3744  
 \insertsubsectionnumber ..... 3733, 3991  
 int commands:  
 \int\_case:nn ..... 381, 4372  
 \int\_compare:nNnTF .....  
 . 546, 910, 2474, 2517, 4050, 4066, 4082, 4172, 4198, 5068  
 \int\_const:Nn ..... 1270, 1872, 1873  
 \int\_div\_truncate:nn ..... 92, 553  
 \int\_eval:n ..... 1864, 1865, 1868, 1869, 3766  
 \int\_from\_hex:n ..... 545  
 \int\_gincr:N ..... 1426, 1727

\int\_gset:Nn .....  
 ... 83, 153, 167, 168, 169, 270, 272, 340, 344, 4173, 4199  
 \int\_if\_odd:nTF ..... 6089  
 \int\_incr:N ..... 3317  
 \int\_mod:nn ..... 555  
 \int\_new:N ..... 56, 82, 152, 1730  
 \int\_set:Nn ..... 535, 544, 2471, 2512, 2769  
 \int\_sub:Nn ..... 549, 2518  
 \int\_to\_Hex:n ..... 547, 552, 554  
 \int\_use:N ..... 1428, 1728, 3321, 5848  
 \int\_value:w .....  
 . 593, 604, 605, 620, 621, 622, 637, 638, 639, 640, 674, 680  
 \int\_zero:N ..... 3314  
 \c\_max\_char\_int ..... 1873  
 \c\_zero\_int ..... 2474  
 \interlinepenalty . 2891, 2933, 2953, 2974, 3098, 3166, 3204

ior commands:

\ior\_close:N ..... 5036  
 \ior\_map\_break: ..... 5029  
 \ior\_map\_inline:Nn ..... 5021  
 \ior\_new:N ..... 5015  
 \ior\_open:Nn ..... 5019

iow commands:

\iow\_close:N ..... 5833  
 \iow\_indent:n ..... 6060  
 \iow\_new:N ..... 5835  
 \iow\_now:Nn ..... 5850  
 \iow\_open:Nn ..... 5830  
 \itemsep ..... 4395, 4406, 4434, 4445  
 \itshape ..... 1929, 1930, 1934, 1935

J

\jfam ..... 1200, 1271, 1926

K

\kaishu ..... 5452  
 \kanjifamily ..... 1920, 2023, 2024, 2025  
 \kanjifamilydefault ..... 2026  
 \kern ..... 1910  
 kernel internal commands:  
 \l\_kernel\_expl\_bool .... 6076, 6081, 6082, 6090, 6091  
 keys commands:  
 \keys\_define:nn 61, 64, 1245, 1275, 1614, 1724, 1878, 2355  
 \keys\_if\_exist:nnTF ..... 1876  
 \l\_keys\_key\_str ..... 129, 131, 147  
 \keys\_set:nn ..... 66, 68  
 \keys\_set\_known:nn ..... 4130  
 \keys\_set\_known:nnN ..... 1287

L

\labelformat ..... 3802, 3838  
 \language ..... 1906  
 \languagealias ..... 4159, 4164, 4184, 4189  
 \LARGE ..... 4413, 4452  
 \Large ..... 2895, 3875, 3881, 3949, 3953, 4412, 4451  
 \large ..... 3974, 4411, 4450  
 \lastbox ..... 3231

`\leavevmode` ..... 1902, 2834, 3113  
`\leftmargin` ..... 4392, 4403, 4431, 4442  
`\leftmargini` ..... 4392, 4403, 4431, 4442  
**legacy commands:**  
`\legacy_if:nTF` . 1156, 3371, 3514, 3603, 3628, 3653, 3678  
`\let` ..... 411,  
 414, 1904, 3146, 3155, 3508, 3510, 4385, 4424, 5514, 5520  
`\linespread` ..... 4460  
`linespread` ..... 12, 177  
`linestretch` ..... 12, 2436  
`\linewidth` ..... 2472, 2480, 2511  
`\lishu` ..... 5452  
`\listfigurename` ..... 2597  
`listfigurename` ..... 13  
`\listtablename` ..... 2598  
`listtablename` ..... 14  
`\LoadClass` ..... 124  
`\long` ..... 5528  
`\ltjdefcharrange` . 850, 851, 852, 854, 859, 860, 864, 867, 868  
`LTJFONTUID` ..... 1724  
`\ltjgetparameter` ..... 2149, 2178, 2278, 2292  
`\ltjlineendcomment` ..... 848  
`\ltjsetkanjiskip` ..... 1913  
`\ltjsetmathletter` ..... 1210  
`\ltjsetparameter` ..... 870, 873, 879, 1897  
`\ltjsetxkanjiskip` ..... 1915  
**lua commands:**  
`\lua_now:n` ..... 967, 2302, 5041, 5201

**M**

`\makeatletter` ..... 935, 939  
`\MAKESPA` ..... 5836, 5906  
`\markboth` ..... 2807, 2901, 2930  
`\mathgroup` ..... 1200  
`\mcfamily` ..... 1934  
`\mdefault` ..... 1257, 1261, 1267, 1602, 1607

**mode commands:**

`\mode_if_math:TF` ..... 1194  
`\mode_leave_vertical:` ..... 55, 685, 691, 697, 703

**msg commands:**

`\msg_critical:nnn` ..... 33, 834, 4485  
`\msg_error:nn` ..... 4565  
`\msg_error:nnn` .....  
 ..... 22, 825, 844, 847, 1508, 2584, 4214, 4294, 4368  
`\msg_error:nnnn` ..... 4531, 4560, 6110  
`\msg_fatal:nn` ..... 5818  
`\g_msg_module_name_prop` ..... 6, 7, 8, 9, 10, 11  
`\g_msg_module_type_prop` ..... 5  
`\msg_new:nnn` ..... 128, 130, 132,  
 146, 148, 371, 828, 1328, 1394, 1476, 1501, 2615, 4104,  
 4486, 4539, 4574, 5011, 5816, 6056, 6123, 6137, 6151, 6247  
`\msg_new:nnnn` . . . 14, 25, 88, 2586, 4216, 4323, 4541, 4568  
`\msg_redirect_name:nnn` ..... 837, 839  
`\msg_warning:nn` .....  
 ..... 220, 376, 1471, 2338, 2349, 2372, 2637, 5064  
`\msg_warning:nnn` 135, 1389, 4123, 4552, 6062, 6150, 6246

`\msg_warning:nnnn` ..... 143, 1324, 4521, 6119, 6136

**N**

`\newCJKfontfamily` ..... 1495  
`\NewDocumentCommand` .....  
 ..... 3, 1355, 1490, 1495, 1503, 1518, 1524, 1530,  
 1537, 1543, 1549, 1555, 1752, 1852, 1919, 2379, 2382,  
 2392, 2419, 2496, 2533, 2540, 2563, 2565, 2732, 4208,  
 4465, 5453, 5454, 5455, 5456, 5458, 5459, 5461, 5473, 5474  
`\newfontfeature` ..... 1272, 1273, 1274  
`\newpage` ..... 2990, 2995  
`\ngostype` ..... 24  
`\nobreak` ..... 2895,  
 2904, 2961, 3078, 3083, 3087, 3102, 3106, 3219, 3876, 3925  
`\nobreakspace` ..... 2895, 2937  
`nocap` ..... 36, 288  
`\noexpand` ..... 5512, 5514, 5515, 5516  
`nofonts` ..... 36  
`noindent` ..... 35  
`nopunct` ..... 36  
`\normalem` ..... 2118  
`\normalfont` ..... 1534, 1541, 1547, 1585,  
 2007, 2029, 2767, 2891, 2933, 2953, 2974, 3073, 3098, 3325  
`\normalsize` ..... 4008, 4025, 4039, 4379, 4418, 4456, 4464  
`nospace` ..... 36  
`nozhmap` ..... 36  
`\null` ..... 2864, 2993  
`\numberline` ..... 3027, 3174, 3188, 3303, 3311, 3499, 3575

**O**

`\onecolumn` ..... 2859

**P**

`\pagestyle` ..... 4150  
`\par` ..... 2794, 2835, 2895, 2901, 2902, 2937,  
 2941, 2943, 2957, 2959, 2978, 2980, 3078, 3083, 3085,  
 3102, 3104, 3114, 3169, 3207, 3219, 3696, 3697, 3705,  
 3708, 3715, 3716, 3724, 3727, 3734, 3735, 3743, 3746,  
 3854, 3876, 3887, 3905, 3913, 3925, 3965, 3967, 3999, 4001  
`paragraph/afterskip` ..... 37  
`paragraph/beforeskip` ..... 37  
`paragraph/numbering` ..... 17  
`paragraph/runin` ..... 23  
`\parindent` ..... 2492, 2493,  
 2515, 2536, 2538, 2543, 2544, 2891, 2953, 3073, 3098, 4084  
`\parsep` ..... 4394, 4395, 4405, 4406, 4433, 4434, 4444, 4445  
`\part` ..... 2832  
`part/beforeskip` ..... 37  
`part/fixbeforeskip` ..... 37  
`part/numbering` ..... 17  
`part/pagestyle` ..... 23, 2742  
`\partmark` ..... 2806, 2889, 2931  
`\partname` ..... 2895, 2937, 3695, 3859  
`\PassOptionsToClass` ..... 121  
`\PassOptionsToPackage` ..... 506, 2068, 2074, 2546, 6064  
`patch/cleveref` ..... 35, 3804  
`\patchcmd` ..... 1910

pdf commands:

`\pdf_object_ref_last`: ..... 484  
`\pdf_object_unnamed_write:nn` ..... 477  
`\pdffontattr` ..... 564, 567  
`\pdfmapline` ..... 5599, 5600, 5601, 5602, 5603, 5604, 5605, 5606, 5607, 5608, 5609, 5610, 5611, 5612, 5613, 5614, 5615, 5616, 5617, 5618, 5619, 5620, 5621, 5622, 5623, 5624, 5656, 5657, 5658, 5659, 5660, 5661, 5662, 5663, 5664, 5665, 5666, 5667, 5668, 5669, 5670, 5671, 5672, 5673, 5753, 5754, 5755, 5756, 5757, 5758, 5759, 5760, 5761, 5762, 5763, 5764, 5765, 5766, 5767, 5768, 5769, 5770, 5771, 5772, 5773, 5774, 5775, 5776, 5777, 5778  
`\pdfnoblue` ..... 482, 566  
`\pdfoutput` ..... 406, 5523  
`\pdfstringdefDisableCommands` ..... 4468  
`\pingfang` ..... 5452  
prg commands:  
`\prg_do_nothing`: ..... 441, 487, 1204, 1702, 1787, 2207, 2325, 3287, 6368  
`\prg_generate_conditional_variant:Nnn` ..... 1381  
`\prg_new_conditional:Npnn` ..... 1118, 2171  
`\prg_new_protected_conditional:Nnn` ..... 5857  
`\prg_new_protected_conditional:Npnn` .... 1368, 6338  
`\prg_return_false`: ..... 1121, 1378, 2175, 2179, 2183, 5877, 6348  
`\prg_return_true`: ..... 1120, 1371, 1376, 2175, 2179, 2183, 5877, 6353  
`\primitive` ..... 5965  
`\ProcessKeyOptions` ..... 363, 364  
`\ProcessKeysOptions` ..... 367  
`\proofname` ..... 2589, 2604  
proofname ..... 14  
prop commands:  
`\prop_clear:N` ..... 1285, 4236  
`\prop_const_from_keyval:Nn` ..... 3769, 4318  
`\prop_get:NnN` ..... 1463  
`\prop_get:NnNTF` ..... 1233, 1370, 1422, 1460, 1695, 1741, 3764, 4212, 4339, 4366  
`\prop_gpop:NnNTF` ..... 1313, 1317, 1822  
`\prop_gput:Nnn` ..... 5, 6, 7, 8, 9, 10, 11, 1291, 1292, 1338, 1450, 1810, 1846  
`\prop_gremove:Nn` ..... 1322  
`\prop_if_empty:NTF` ..... 1384, 1566, 1646  
`\prop_map_break:n` ..... 1580  
`\prop_map_inline:Nn` ..... 1578, 1654  
`\prop_new:N` ..... 1237, 1308, 1309, 1310, 1455, 1636, 1830, 1851, 4223  
`\prop_put:Nnn` ..... 1633, 4227  
`\prop_remove:Nn` ..... 1632  
`\prop_set_from_keyval:Nn` ..... 1238  
`\prop_to_keyval:N` ..... 4319  
`\protect` ..... 596, 610, 627, 2820, 2826, 3027, 3048, 3049, 3174, 3188, 3303, 3311, 3498, 3521, 3523, 3527, 3529, 3533, 3535, 3543, 3548, 3560, 3561, 6299, 6300  
`\ProvideDocumentCommand` ..... 2806  
`\ProvidesFile` . 5526, 5535, 5537, 5539, 5541, 5543, 5545, 5547

`\providetranslation` ..... 5970, 5971, 5972, 5973, 5974, 5975, 5976, 5977, 5978, 5979, 5980, 5981, 5982, 5983, 5984, 5985, 5986, 5987, 5988, 5989, 5990, 5991, 5992, 5993, 5994, 5995, 5996, 5997, 5998, 5999, 6000, 6001, 6002, 6003, 6004, 6005, 6006, 6007, 6008, 6009, 6010, 6011, 6012, 6013, 6014, 6015, 6016, 6017, 6018, 6019, 6020, 6021, 6022, 6023, 6024, 6025, 6026, 6027, 6028, 6029, 6030, 6031, 6032, 6033, 6034, 6035, 6036, 6037, 6038, 6039, 6040, 6041, 6042, 6043, 6044, 6045, 6046, 6047, 6048, 6049  
punct ..... 11, 36, 254, 2341  
`\punctstyle` ..... 575, 2346

Q

`\quad` . 3139, 3606, 3607, 3609, 3610, 3614, 3615, 3627, 3632, 3639, 3642, 3645, 3648, 3656, 3657, 3659, 3660, 3664, 3665, 3677, 3682, 3882, 3933, 3942, 3975, 4009, 4026, 4040  
quark commands:  
`\q_mark` ..... 1655, 1663, 6342, 6345  
`\q_stop` ..... 105, 108, 1155, 1158, 1161, 1164, 5058, 5315, 6307, 6312, 6321, 6342, 6345  
`\quotation` ..... 4478

R

`\raggedright` ..... 2891, 2953, 3073, 3098, 3874, 3923  
`\refname` ..... 2605, 2610, 2617, 2626, 2634, 2635  
refname ..... 14  
`\refstepcounter` ... 2875, 2916, 3023, 3150, 3784, 3841, 3842  
regex commands:  
`\regex_replace_once:nnN` ..... 5026  
`\relax` ..... 401, 404, 405, 408, 1902, 2913, 2936, 3115, 3151, 3159, 3165, 3181, 3197, 3210, 3216, 3235, 3638, 3641, 3697, 3708, 3716, 3727, 3735, 3746, 5485, 5490, 5493, 5521, 5522, 5527  
`\renewcommand` ..... 2833, 2849, 3003  
`\RenewDocumentCommand` ..... 3842  
`\RequirePackage` ..... 4, 36, 37, 51, 297, 303, 314, 324, 366, 509, 513, 514, 515, 818, 842, 845, 959, 964, 1990, 2079, 2089, 2549, 4203  
`\RequirePackageWithOptions` ..... 6065  
resetalternatefont ..... 33, 1793  
reverse commands:  
`\reverse_if:N` ..... 1993  
`\rmdefault` ..... 2041  
`\rmfamily` ..... 2004

S

scan commands:  
`\scan_stop`: ..... 78, 105, 923, 1090, 1200, 1788, 2461, 2472, 5841, 5842, 6186  
scheme ..... 11, 276  
`\scriptsize` ..... 4409, 4448  
`\secdef` ..... 2846, 2868, 3015  
secnumdepth ..... 17, 3753  
section/afterskip ..... 37  
section/beforeskip ..... 37  
section/numbering ..... 17

section/runin .....	23	\skip_sub:Nn .....	2789, 2796
\sectionname .....	3714, 3959	\skip_use:N .....	2820, 2826
\selectfont .....	1362, 1813, 1826, 1920, 2004, 2005, 2006, 2242, 2249, 2271, 2421, 4213	\skip_zero:N .....	2485
seq commands:		\c_zero_skip .....	2701, 2702, 2750, 2751, 2817, 2823
\seq_clear:N .....	4237	\small .....	4387, 4426
\seq_const_from_clist:Nn .....	448, 2646, 2649, 3360, 4320, 5820	\songti .....	5452
\seq_gpop:NN .....	6088	\space .....	406, 3024, 3078, 3859, 3921, 3959, 3990, 4091
\seq_gpop:NNTF .....	1066	space .....	11, 36, 259, 2320
\seq_gpush:Nn .....	1061, 6080	\special .....	1956, 1957, 1960, 1961, 1969, 1970, 1973, 1974, 2087, 2111, 5553, 5554, 5555, 5556, 5557, 5558, 5559, 5560, 5561, 5562, 5563, 5564, 5565, 5566, 5567, 5568, 5569, 5570, 5577, 5578, 5579, 5580, 5581, 5582, 5583, 5584, 5585, 5586, 5587, 5588, 5589, 5590, 5591, 5592, 5593, 5594, 5626, 5627, 5628, 5629, 5630, 5631, 5632, 5633, 5634, 5635, 5636, 5637, 5638, 5639, 5640, 5641, 5642, 5643, 5644, 5645, 5646, 5647, 5648, 5649, 5650, 5651, 5675, 5676, 5677, 5678, 5679, 5680, 5681, 5682, 5683, 5684, 5685, 5686, 5687, 5688, 5689, 5690, 5691, 5692, 5699, 5700, 5701, 5702, 5703, 5704, 5705, 5706, 5707, 5708, 5709, 5710, 5711, 5712, 5713, 5714, 5715, 5716, 5717, 5718, 5719, 5720, 5721, 5722, 5723, 5724, 5731, 5732, 5733, 5734, 5735, 5736, 5737, 5738, 5739, 5740, 5741, 5742, 5743, 5744, 5745, 5746, 5747, 5748, 5780, 5781, 5782, 5783, 5784, 5785, 5786, 5787, 5788, 5789, 5790, 5791, 5792, 5793, 5794, 5795, 5796, 5797, 5798, 5799, 5800, 5801, 5802, 5803, 5804, 5805
\seq_gput_right:Nn .....	1388	\SplitArgument .....	1753, 1853, 2733
\seq_if_in:NnTF .....	454, 1386	str commands:	
\seq_map_function:NN .....	3431, 3564	\c_backslash_str .....	6326
\seq_map_inline:Nn .....	2759, 3315, 3363, 3374, 3384, 3407, 3490, 3795, 5845	\c_colon_str .....	1187, 1188, 1189
\seq_new:N .....	1069, 1393, 4224, 6094	\c_dollar_str .....	5380
\seq_put_right:Nn .....	4232	\c_left_brace_str .....	6327
\seq_use:Nn .....	4321	\str_case:nnTF .....	2039, 4238, 6297, 6312
\seq_use:Nnnn .....	4221	\str_case_e:nnTF .....	1401, 1412
\setbox .....	3231	\str_const:Nn .....	23, 30
\setCJKfamilyfont .....	1407, 1495, 4763, 4765, 4767, 4769, 4805, 4806, 4807, 4808, 4827, 4829, 4831, 4833, 4876, 4878, 4880, 4882, 4895, 4896, 4897, 4898, 4899, 4900, 4941, 4943, 4944, 4945, 4946, 4947, 4960, 4962, 4964, 4965, 5001, 5003, 5005, 5006, 5096, 5097, 5098, 5099, 5100, 5101, 5144, 5145, 5146, 5147, 5166, 5168, 5173, 5180, 5185, 5189, 5192, 5264, 5266, 5278, 5287, 5292, 5302, 5307, 5330, 5332, 5334, 5372, 5374, 5376, 5397, 5398, 5399, 5400, 5401, 5403, 5404, 5443, 5444, 5445, 5446, 5447, 5448, 5449	\str_if_empty:NnTF .....	6383, 6419, 6428
\setCJKmainfont .....	1403, 1530, 4753, 4801, 4817, 4866, 4891, 4937, 4956, 4997, 5088, 5140, 5155, 5163, 5251, 5255, 5324, 5367, 5392, 5440	\str_if_eq:nnTF .....	512, 574, 581, 647, 1570, 2075, 2084, 4155, 4181, 4195, 4551, 4554, 4571, 4576, 6303
\setCJKmathfont .....	1530	\str_if_eq_p:nn .....	4518, 4519
\setCJKmonofont .....	1405, 1530, 4761, 4804, 4825, 4874, 4894, 4940, 4959, 5000, 5095, 5143, 5179, 5182, 5283, 5285, 5328, 5371, 5396, 5442	\str_lowercase:n .....	475
\setCJKromanfont .....	1536, 1562	\str_new:N .....	6332, 6333, 6334
\setCJKsansfont .....	1404, 1530, 4759, 4803, 4823, 4872, 4893, 4939, 4958, 4999, 5094, 5142, 5172, 5176, 5269, 5271, 5326, 5369, 5394, 5441	\str_set:Nn .....	6350, 6351, 6352
\setcounter .....	3765, 3766	\strutbox .....	4462
\SetSymbolFont .....	1268, 1606, 1610, 1925	sub3section .....	11, 268
\sfdefault .....	2042	sub4section .....	11, 268
\sffamily .....	2005	subparagraph/afterskip .....	37
\shapedefault .....	1257, 1261, 1263, 1267, 1269, 1602, 1604, 1607, 1611	subparagraph/beforeskip .....	37
skip commands:		subparagraph/numbering .....	17
\skip_horizontal:N .....	2159, 3273	subparagraph/runin .....	23
\skip_horizontal:n .....	2359	subsection/afterskip .....	37
\skip_if_eq:nnTF .....	2178, 2182, 2277, 2817, 2823	subsection/beforeskip .....	37
\skip_new:N .....	2170, 2290, 2798	subsection/numbering .....	17
\skip_set:Nn .....	2149, 2152, 2281, 2291, 2365, 2462, 2476, 2520, 2528, 2800, 2816, 2822, 3271	subsection/runin .....	23
\skip_set_eq:NN .....	2168, 2283, 2367	\subsectionname .....	3733, 3990

subsubsection/numbering ..... 17  
 subsubsection/runin ..... 23  
 \symmincho ..... 1926  
**sys commands:**  
 \c\_sys\_engine\_str ..... 24  
 \sys\_get\_shell:nnN ..... 5869, 5871  
 \sys\_if\_engine luatex:TF ..... 5013, 5151  
 \sys\_if\_engine pdftex:TF ... 217, 300, 375, 4490, 4496  
 \sys\_if\_engine pdftex\_p: ..... 5464  
 \sys\_if\_engine uptex:TF ..... 4492, 4498  
 \sys\_if\_engine uptex\_p: ..... 5465  
 \sys\_if\_engine xetex:TF ..... 294, 5814  
 \sys\_if\_output\_pdf:TF ..... 94, 490, 2078, 4497  
 \sys\_if\_platform\_windows:TF ..... 4502

**T**

\tablename ..... 2600  
 tablename ..... 14  
 \TeX ..... 4466

**T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> commands:**

\@@ifdefinable ..... 900  
 \@@italiccorr ..... 1911  
 \@@par ..... 3166, 3943, 3976, 4010, 4054, 4070  
 \@afterheading ..... 2909, 2966, 3055, 3065, 3224  
 \@afterindentfalse ..... 2844, 3009, 3013, 3118, 3124  
 \@afterindenttrue ..... 2842, 3011, 3116, 3122  
 \@Alph ..... 4087, 4092  
 \@break@tfor ..... 414  
 \@chapapp ..... 3024, 3078, 3623, 3644, 3673  
 \@chapter ..... 3015, 3016  
 \@chinese ..... 2551, 2560  
 \@classoptionslist ..... 49, 409, 415, 417  
 \@clubpenalty ..... 3239  
 \@currenthref ..... 3396  
 \@currentlabel ..... 3786, 3790  
 \@currentxt ..... 160, 6055, 6213, 6219  
 \@currname ..... 160, 6055, 6112, 6214, 6220  
 \@dblarg ..... 3137  
 \@defaultfamilyhook ..... 2007  
 \@empty ..... 117, 3146, 3155  
 \@endpart ..... 2945, 2982, 2984  
 \@EverySelectfont@Init ..... 2210, 2225, 2237  
 \@firstofone ..... 63  
 \@firstoftwo ..... 3560  
 \@gobble ..... 63  
 \@hangfrom ..... 3165, 3203  
 \@ifclassloaded ..... 4100  
 \@ifdefinable ..... 900  
 \@ifl@t@r ..... 6069  
 \@ifpackagelater ..... 21, 824, 843, 846, 872, 2125  
 \@ifpackageloaded ..... 833,  
 954, 2061, 2072, 2099, 3470, 3586, 3813, 6109, 6145, 6158  
 \@ifpackagewith ..... 3415, 3424, 3475, 3650, 3815  
 \@ifstar ..... 1907, 3135  
 \@listI ..... 4385, 4424  
 \@listi ..... 4385, 4392, 4403, 4424, 4431, 4442

\@M ..... 2891, 2933, 2953, 2974, 3098, 3166, 3204, 3232  
 \@makechapterhead ..... 3052, 3054, 3067  
 \@makeother ..... 1904  
 \@makeschapterhead ..... 3062, 3064, 3092  
 \@minus ..... 3944, 3977, 4011, 4027, 4041, 4381,  
 4383, 4389, 4391, 4393, 4394, 4400, 4402, 4404, 4405,  
 4420, 4422, 4428, 4430, 4432, 4433, 4439, 4441, 4443, 4444  
 \@namedef .....  
 ... 1898, 3521, 3523, 3527, 3529, 3533, 3535, 3543, 3548  
 \@nameuse ..... 3521, 3523, 3533, 3535, 3543  
 \@ne ..... 411, 3468, 3609, 3641, 3659  
 \@nil ..... 1141  
 \@nobreakfalse ..... 3226  
 \@noligs ..... 1905  
 \@noskipsecfalse ..... 3230  
 \@noskipsectrue ..... 3227  
 \@onlypreamble ..... 43, 445, 523, 1558,  
 1559, 1560, 1561, 1562, 1563, 1985, 1986, 1987, 1988, 4544  
 \@part ..... 2846, 2868, 2870  
 \@pkgextension ..... 6121, 6130, 6131, 6213, 6219  
 \@plus ..... 3888,  
 3889, 3944, 3945, 3977, 3978, 4011, 4012, 4027, 4041,  
 4055, 4071, 4381, 4382, 4383, 4389, 4390, 4391, 4393,  
 4394, 4400, 4401, 4402, 4404, 4405, 4420, 4421, 4422,  
 4428, 4429, 4430, 4432, 4433, 4439, 4440, 4441, 4443, 4444  
 \@popfilename ..... 160, 6216  
 \@pushfilename ..... 160  
 \@reset@options ..... 160, 6210  
 \@rmfamilyhook ..... 2004  
 \@schapter ..... 3015, 3057  
 \@secCNTformat ..... 3138, 3151  
 \@secondoftwo ..... 3561  
 \@secpenalty ..... 3129, 3947, 3980, 4014, 4028, 4042  
 \@sect ..... 3137, 3142  
 \@setfontsize ..... 4371  
 \@sffamilyhook ..... 2005  
 \@spart ..... 2846, 2868, 2947  
 \@ssect ..... 3136, 3194  
 \@startsection ..... 107, 110, 119, 3112, 3320  
 \@sverb ..... 1907  
 \@svsec ..... 3146, 3151, 3155, 3165, 3167, 3181, 3182  
 \@svsechd ..... 3179, 3210, 3211, 3233  
 \@tempdima ..... 3569, 3571  
 \@tempskipa ..... 3115, 3117, 3118,  
 3129, 3159, 3160, 3197, 3198, 3216, 3217, 3220, 3235, 3236  
 \@tempswafalse ..... 2862  
 \@tempswatrue ..... 2860  
 \@text@composite@x ..... 60, 950  
 \@tfor ..... 412  
 \@topnewpage ..... 3052, 3062  
 \@topnum ..... 3008  
 \@ttfamilyhook ..... 2006  
 \@verb ..... 1907  
 \@xsect ..... 3193, 3214, 3215  
 \@zhdig ..... 2559  
 \@zhnum ..... 2551, 2558

<code>\abstractname</code> .....	14	<code>\CJKsymbol</code> .....	88
<code>\add@unicode@accent</code> .....	60, 951	<code>\CJKunderdotbasesep</code> .....	36
<code>\addCJKfontfeature</code> .....	78	<code>\CJKunderline</code> .....	36
<code>\AfterEndPreamble</code> .....	159	<code>\clearpage</code> .....	53
<code>\algorithmname</code> .....	14	<code>\contentsname</code> .....	13
<code>\appendixname</code> .....	14, 28	<code>\CS</code> .....	64
<code>\AtBeginDocument</code> .....	43, 53	<code>\CTEX@add</code> .....	411, 414, 415
<code>\AtBeginDvi</code> .....	50	<code>\CTEX@add@unicode@accent</code> .....	919, 951
<code>\AtBeginShipoutFirst</code> .....	50	<code>\CTEX@addloflotskip</code> .....	2813, 3050
<code>\AtEndOfClass</code> .....	43	<code>\CTEX@addtocline</code> ...	2830, 2888, 2929, 3046, 3177, 3191
<code>\AtEndOfPackage</code> .....	43	<code>\CTEX@afterindent</code> .....	3121, 3281, 3288
<code>\AtEndOfPackageFile*</code> .....	159	<code>\CTEX@aftertitle</code> .....	3207, 3212, 3280, 3287
<code>\AtEndPreamble</code> .....	159	<code>\CTEX@alchar</code> .....	888, 911
<code>\baselineskip</code> .....	12, 89, 92	<code>\CTEX@appendix@number</code> .....	3335, 3347, 3353
<code>\bfseries</code> .....	57	<code>\CTEX@appendix@numbering</code> .....	3336, 3349, 3355
<code>\bibname</code> .....	14	<code>\CTEX@beginallalchar</code> .....	888, 915, 921, 936, 960
<code>\c@chapter</code> .....	4092	<code>\CTEX@chapter@afterindent</code> .....	3010
<code>\c@secnumdepth</code> ..	2872, 2894, 2913, 2936, 3017, 3076, 3143, 3173, 3187, 3606, 3609, 3614, 3621, 3622, 3631, 3638, 3641, 3644, 3647, 3656, 3659, 3664, 3671, 3672, 3681	<code>\CTEX@chapter@aftername</code> .....	3082
<code>\c@section</code> .....	4087	<code>\CTEX@chapter@afterskip</code> .....	3089, 3108
<code>\caption</code> .....	37	<code>\CTEX@chapter@aftertitle</code> .....	3085, 3104
<code>\catcode</code> .....	59, 147, 160	<code>\CTEX@chapter@beforeskip</code> .....	3069, 3094
<code>\catcodetable</code> .....	59	<code>\CTEX@chapter@break</code> .....	3006
<code>\ccwd</code> .....	12, 25, 31, 36, 43, 86, 87, 91-93	<code>\CTEX@chapter@fixskip</code> .....	3070, 3090, 3095, 3109
<code>\chapter</code> .....	10, 18, 23, 28, 29, 99, 106, 123	<code>\CTEX@chapter@format</code> .....	3075, 3100
<code>\chaptermark</code> .....	15	<code>\CTEX@chapter@lofskip</code> .....	2748
<code>\chaptername</code> .....	18	<code>\CTEX@chapter@lotskip</code> .....	2749
<code>\char</code> .....	61	<code>\CTEX@chapter@numbering</code> .....	3021, 3355
<code>\chardef@text@cmd</code> .....	60, 948	<code>\CTEX@chapter@pagestyle</code> .....	2747
<code>\CheckCommand</code> .....	87	<code>\CTEX@chapter@titleformat</code> .....	3084, 3103
<code>\Chinese</code> .....	37	<code>\CTEX@chapter@tocline</code> .....	3292
<code>\chinese</code> .....	31, 37, 94	<code>\CTEX@chaptername</code> .....	3082
<code>\CJK@@ignorespaces</code> .....	54, 591	<code>\CTEX@char@n</code> .....	593, 683, 718, 749
<code>\CJK@addcmap</code> .....	561	<code>\CTEX@char@nn</code> .....	56, 602, 674, 683, 707, 723
<code>\CJK@encoding</code> ....	68, 70, 77, 1044, 1048, 1135, 1252, 1306, 1601, 1604, 1606, 1610, 1750, 1760, 1761, 1779, 1780	<code>\CTEX@char@nnn</code> .....	56, 707
<code>\CJK@envStart</code> .....	53, 586	<code>\CTEX@char@nnn</code> .....	618, 683, 708, 724
<code>\CJK@family</code> .....	63, 1035, 1049, 1103, 1135, 1359, 1812, 1822, 1824, 2271, 2272	<code>\CTEX@char@nnnn</code> .....	707
<code>\CJK@ignorespaces</code> .....	57, 687, 693, 699, 705, 756, 760	<code>\CTEX@char@nnnnn</code> .....	635, 683, 709, 725
<code>\CJK@input</code> .....	532	<code>\CTEX@char@nnnnn</code> .....	707
<code>\CJK@loadBinding</code> .....	582, 583	<code>\CTEX@chardef@text@cmd</code> .....	898, 948
<code>\CJK@makeActive</code> .....	53	<code>\CTEX@defaultfamilyhook</code> .....	2017
<code>\CJK@plane</code> .....	52, 88, 462, 463, 2260	<code>\CTEX@document@left@hook</code> .....	6183, 6189
<code>\CJK@surr</code> .....	52, 540	<code>\CTEX@document@right@hook</code> .....	6185, 6191
<code>\CJK@upperReset</code> .....	53	<code>\CTEX@endallalchar</code> .....	888, 917, 924, 940, 961
<code>\CJK@X</code> .....	54, 592	<code>\CTEX@Family@CMap</code> .....	459, 496, 501
<code>\CJK@XX</code> .....	592	<code>\CTEX@fixheadingskip</code> .....	2787, 2792, 2839, 2907, 2964, 2988, 3090, 3109, 3132, 3222
<code>\CJK@XXX</code> .....	592	<code>\CTEX@fixskip</code> .....	3132, 3222, 3282, 3289
<code>\CJK@XXXX</code> .....	592	<code>\CTEX@fixtopskip</code> .....	2784, 2866, 3070, 3095
<code>\CJKfamily</code> .....	70, 81	<code>\CTEX@getttitle</code> .....	2887, 2928, 2951, 2972, 3045, 3060, 3158, 3196, 3402
<code>\CJKfamilydefault</code> .....	53, 74, 83, 84	<code>\CTEX@hang</code> .....	3251, 3283, 3290
<code>\CJKglue</code> .....	92, 129	<code>\CTEX@heading@format@initial</code> .....	2765, 2892, 2934, 2954, 2975, 3074, 3099, 3163, 3201, 3698, 3717, 3736
<code>\CJKhook</code> .....	57	<code>\CTEX@heading@glue</code> .....	3237, 3268
<code>\CJKrmddefault</code> .....	52	<code>\CTEX@headinghang</code> 2897, 2939, 2956, 2977, 3081, 3101, 3244	

<code>\CTEX@headingskip</code> .....	2801, 2840, 2867, 2908, 2965, 2989, 3071, 3091, 3096, 3110, 3133, 3223	<code>\CTEX@setheadingskip</code> .....	2799, 2838, 2865, 2906, 2963, 2987, 3069, 3089, 3094, 3108, 3131, 3221
<code>\CTEX@hyperheadinghook</code> .....	3368, 3400	<code>\CTEX@setthispagestyle</code> .....	2802, 2857, 3007
<code>\CTEX@ifnamefalse</code> ...	2808, 2878, 2883, 2919, 2924, 2949, 2970, 3029, 3035, 3041, 3058, 3144, 3153, 3195, 3514	<code>\CTEX@sffamilyhook</code> .....	2015, 2024
<code>\CTEX@ifnametrue</code> ...	2808, 2874, 2915, 3022, 3149, 3514	<code>\CTEX@subsection@aftername</code> .....	3740
<code>\CTEX@indentbox</code> ...	2782, 3182, 3211, 3700, 3719, 3738	<code>\CTEX@subsection@afterskip</code> .....	3746
<code>\CTEX@makeanchor</code> .....	2879, 2884, 2920, 2925, 3030, 3154, 3358, 3366, 3379, 3385, 3391	<code>\CTEX@subsection@aftertitle</code> .....	3744
<code>\CTEX@makeanchor@chapter</code> .....	3036, 3042	<code>\CTEX@subsection@beforeskip</code> .....	3735
<code>\CTEX@makeanchor@schapter</code> .....	3059	<code>\CTEX@subsection@format</code> .....	3737
<code>\CTEX@makeanchor@sect</code> .....	3145	<code>\CTEX@subsection@indent</code> .....	3738
<code>\CTEX@makeanchor@spart</code> .....	2950, 2971	<code>\CTEX@subsection@numbering</code> .....	3739
<code>\CTEX@makeanchor@ssect</code> .....	3136	<code>\CTEX@subsection@titleformat</code> .....	3744
<code>\CTEX@part@afterindent</code> .....	2841	<code>\CTEX@subsectionname</code> .....	3740
<code>\CTEX@part@aftername</code> .....	2898, 2940, 3702	<code>\CTEX@text@composite@x</code> .....	913, 950
<code>\CTEX@part@afterskip</code> .....	2906, 2963, 2987, 3708	<code>\CTEX@textcmd</code> .....	933, 934, 938, 941
<code>\CTEX@part@aftertitle</code> ...	2902, 2943, 2959, 2980, 3706	<code>\CTEX@thechapter</code> .....	3353
<code>\CTEX@part@beforeskip</code> .....	2838, 2865, 3697	<code>\CTEX@thesection</code> .....	3347
<code>\CTEX@part@break</code> .....	2836, 2855	<code>\CTEX@titleformat@n</code> .....	107, 3206, 3212, 3279, 3286
<code>\CTEX@part@fixskip</code> .....	2839, 2866, 2907, 2964, 2988	<code>\CTEX@titlesplabel@clear</code> .....	3530, 3540
<code>\CTEX@part@format</code> .....	2893, 2935, 2955, 2976, 3699	<code>\CTEX@titlesplabel@set</code> .....	3524, 3536, 3540
<code>\CTEX@part@indent</code> .....	3700	<code>\CTEX@toc@width@n</code> .....	3566, 3581, 3593
<code>\CTEX@part@numbering</code> .....	2873, 2914, 3701	<code>\CTEX@todayold</code> .....	2567, 2572
<code>\CTEX@part@pagestyle</code> .....	2746	<code>\CTEX@ttfamilyhook</code> .....	2016, 2025
<code>\CTEX@part@titleformat</code> ..	2900, 2942, 2958, 2979, 3706	<code>\CTEX@update@sectionformat@n</code> .....	3120, 3277
<code>\CTEX@part@tocline</code> .....	3292	<code>\CTEX@updatettlifname</code> .....	3511, 3513
<code>\CTEX@partname</code> .....	2898, 2940, 3702	<code>\CTEX@verbatim@font@hook</code> .....	1892, 1896
<code>\CTEX@patch@text@cmd</code> .....	926, 952	<code>\ctex@zhmap@endinput</code> .....	5808
<code>\CTEX@patch@tunec</code> .....	946	<code>\CTEX@digits</code> .....	31
<code>\CTEX@popfilename@hook</code> .....	6216	<code>\CTEX@ifname</code> .....	27, 29, 111
<code>\CTEX@postappendix</code> .....	3340, 3348, 3354	<code>\CTEX@number</code> .....	31
<code>\CTEX@postchapter</code> .....	3354	<code>\CTEX@numberline</code> .....	27
<code>\CTEX@postsection</code> .....	3348	<code>\CTEX@options</code> .....	37
<code>\CTEX@preappendix</code> .....	3339, 3346, 3352	<code>\ctexset</code> .....	5, 6, 8, 9, 13, 16, 17, 32, 33, 35, 37, 131
<code>\CTEX@prechapter</code> .....	3352	<code>\CTEX@setup</code> .....	37
<code>\CTEX@presection</code> .....	3346	<code>\CTEX@thechapter</code> .....	15, 28, 111
<code>\CTEX@reset@options@hook</code> .....	6210	<code>\CTEX@thesection</code> .....	15, 19, 114
<code>\CTEX@rmfamilyhook</code> .....	83, 1994, 2014, 2023	<code>\CTEX@underdotbasesep</code> .....	36
<code>\CTEX@runin</code> .....	3161, 3199, 3218, 3284, 3291	<code>\CTEX@underline</code> .....	36
<code>\CTEX@save@appendix</code> .....	3341, 3344	<code>\curr@fontshape</code> .....	1051, 1085, 1094, 1100, 1142
<code>\CTEX@save@refstepcounter</code> .....	3841, 3844	<code>\declare@file@substitution</code> .....	158, 6129
<code>\CTEX@section@aftername</code> .....	3721	<code>\DeclareAlternateKanjiFont</code> .....	77
<code>\CTEX@section@afterskip</code> .....	3727	<code>\DeclareFontEncoding</code> .....	66
<code>\CTEX@section@aftertitle</code> .....	3725	<code>\DeclareFontFamily</code> .....	50, 65
<code>\CTEX@section@beforeskip</code> .....	3716	<code>\DeclareFontShape</code> .....	65
<code>\CTEX@section@format</code> .....	3718	<code>\DeclareFontShape@</code> .....	77, 1711
<code>\CTEX@section@indent</code> .....	3719	<code>\DeclareRobustCommand</code> .....	161
<code>\CTEX@section@numbering</code> .....	3349, 3720	<code>\DeclareTextCommand</code> .....	60
<code>\CTEX@section@titleformat</code> .....	3725	<code>\DeclareTextSymbol</code> .....	60
<code>\CTEX@sectionhang</code> .....	3167, 3205, 3244	<code>\DeclareUnicodeAccent</code> .....	60
<code>\CTEX@sectionname</code> .....	3721	<code>\DeclareUnicodeComposite</code> .....	60
<code>\CTEX@selectfont@hook</code> .....	2202, 2260, 2263	<code>\define@newfont</code> .....	64, 1077
<code>\CTEX@selectfont@save</code> .....	2216, 2231, 2235, 2238	<code>\disable@package@load</code> .....	158, 6116, 6118
<code>\CTEX@setcurrentlabel@n</code> .....	3784, 3845	<code>\do@noligs</code> .....	1910
		<code>\do@subst@correction</code> .....	64, 65, 1076
		<code>\em</code> .....	81
		<code>\endlinechar</code> .....	52

<code>\escapechar</code> .....	67	<code>\insertcontinuationtext</code> .....	14
<code>\EverySelectfont</code> .....	87	<code>\itdefault</code> .....	65
<code>\expandafter</code> .....	35	<code>\itshape</code> .....	58
<code>\external@font</code> .....	1087, 1088, 1090	<code>\jfam</code> .....	81
<code>\extract@font</code> .....	64, 1075	<code>\kaishu</code> .....	8
<code>\f@baselineskip</code> .....	2461	<code>\kanjifamilydefault</code> .....	83
<code>\f@encoding</code> .....	1048, 1093, 1197	<code>\kanjiskip</code> .....	89
<code>\f@family</code> .....	63, 1049, 1093, 1103	<code>\l@nohyphenation</code> .....	1906
<code>\f@nch@initialise</code> .....	3635	<code>\labelformat</code> .....	34, 35, 117, 118
<code>\f@series</code> .....	1044, 1135	<code>\leftmark</code> .....	15
<code>\f@shape</code> .....	1044, 1135	<code>\linewidth</code> .....	92
<code>\f@size</code> 1044, 1051, 1130, 1138, 1141, 1142, 2145, 2199, 3572		<code>\lishu</code> .....	8
<code>\fangsong</code> .....	8	<code>\listfigurename</code> .....	13
<code>\fbox</code> .....	20	<code>\listtablename</code> .....	14
<code>\figurename</code> .....	14	<code>\ltj@@does@alt@set</code> .....	65
<code>\font</code> .....	65	<code>\ltj@@getjfontnumber</code> .....	66, 1150
<code>\font@name</code> 63, 65, 564, 566, 567, 1054, 1061, 1062, 1067, 1072, 1090, 1091, 1110, 1178, 1179, 1180, 1181, 1182, 1189		<code>\ltj@@set@stackfont</code> .....	1187, 1188, 1189
<code>\fontcharwd</code> .....	87	<code>\ltj@allalchar</code> .....	894
<code>\footnotesep</code> .....	129	<code>\ltj@curjfmt</code> .....	64, 88, 1097
<code>\footnotesize</code> .....	129	<code>\ltj@pickup@altfont@auxy</code> .....	63, 1136
<code>\get@external@font</code> .....	1084	<code>\ltj@pickup@altfont@copy</code> .....	63, 1148
<code>\getanddefine@fonts</code> .....	1167	<code>\ltj@setpar@global</code> .....	80, 1186, 1913, 1915
<code>\globaldefs</code> .....	63	<code>\ltj@tempcntc</code> .....	66, 1151
<code>\globaljfont</code> .....	64	<code>\ltjalchar</code> .....	61
<code>\H@old@chapter</code> .....	3373	<code>\ltjlineendcomment</code> .....	59
<code>\H@refstepcounter</code> .....	3817	<code>\ltjsetkanjiskip</code> .....	80
<code>\hbox</code> .....	24, 99	<code>\ltjsetparameter</code> .....	80
<code>\heiti</code> .....	8, 146	<code>\ltjsetxkanjiskip</code> .....	80
<code>\hrule</code> .....	99	<code>\m@ne</code> .....	2872, 2894, 3017, 3076, 3614, 3621, 3622, 3644, 3671, 3672, 4596, 4597, 4598, 4599, 4600, 4601
<code>\Hy@chapapp</code> .....	3030, 3036, 3042, 3059	<code>\MakeUppercase</code> .....	54
<code>\Hy@driver</code> .....	2084	<code>\markboth</code> .....	15
<code>\Hy@MakeCurrentHrefAuto</code> .....	3393	<code>\markright</code> .....	15
<code>\Hy@org@chapter</code> .....	3373	<code>\math@bgroup</code> .....	1196
<code>\Hy@raisedlink</code> .....	3394	<code>\math@egroup</code> .....	1201
<code>\hyper@anchorend</code> .....	3397	<code>\math@fonts</code> .....	1184
<code>\hyper@anchorstart</code> .....	3396	<code>\mathbb</code> .....	129
<code>\hypersetup</code> .....	84	<code>\maxdimen</code> .....	12, 92
<code>\if@compatibility</code> .....	1899	<code>\meaning</code> .....	161–163
<code>\if@mainmatter</code> .....	3019, 3077, 3621, 3671, 4118, 4119	<code>\mule@arg</code> .....	671
<code>\if@nobreak</code> .....	3126	<code>\newcommand</code> .....	161
<code>\if@noskipsec</code> .....	2834, 3113, 3229	<code>\newrobustcmd</code> .....	161
<code>\if@openright</code> .....	2850, 2992, 3004, 3891, 3919	<code>\newtitlemark</code> .....	112, 113
<code>\if@tempswa</code> .....	2998	<code>\nfss@catcodes</code> .....	68
<code>\if@twocolumn</code> .....	2858, 3051, 3061	<code>\noindent</code> .....	98
<code>\if@twoside</code> .....	2991	<code>\normalfont</code> .....	83
<code>\iffalse</code> .....	66	<code>\normalsize</code> .....	10, 124
<code>\IfFormatAtLeastTF</code> .....	157	<code>\nouppercase</code> .....	52
<code>\ifin@</code> .....	66, 414	<code>\NR@getttitle</code> .....	3406
<code>\iftrue</code> .....	66	<code>\null</code> .....	80
<code>\ifttl@label</code> .....	3508, 3510	<code>\numberline</code> .....	113
<code>\ifttl@toclabel</code> .....	3508, 3510	<code>\p@</code> .....	2145, 2199, 2937, 3048, 3049, 3068, 3078, 3088, 3093, 3107, 3572, 3887, 3915, 3916, 3917, 3918, 3925, 4381, 4382, 4383, 4389, 4390, 4391, 4393, 4394, 4400, 4401, 4402, 4404, 4405, 4420, 4421, 4422, 4428, 4429, 4430, 4432, 4433, 4439, 4440, 4441, 4443, 4444
<code>\in@</code> .....	413		
<code>\indexname</code> .....	14		
<code>\input</code> .....	157		
<code>\InputIfFileExists</code> .....	157		

- \pagenumbering ..... 31
- \pagestyle ..... 11
- \par@update ..... 88, 2222, 2226, 2230
- \paragraph ..... 11, 23, 42
- \parindent ..... 12, 36, 91, 93, 98, 107, 129
- \parskip ..... 99
- \part ..... 17, 23, 25, 26, 99, 106
- \partmark ..... 99
- \partname ..... 18
- \partpage ..... 16, 115
- \PassOptionsToClass ..... 43
- \PassOptionsToPackage ..... 84, 94
- \pdfmapline ..... 135
- \pickup@font ..... 63, 1050
- \pickup@jfont ..... 1081
- \pingfang ..... 8, 146
- \ProcessKeysOptions ..... 34, 43
- \proofname ..... 14, 95
- \protected ..... 54, 112
- \protected@edef ..... 2564, 2566, 3151, 3786
- \ProvidesFile ..... 148
- \ps@headings ..... 114, 3602, 3655, 3658, 3663, 3669, 3680
- \refname ..... 14
- \refstepcounter ..... 35, 117
- \refstepcounter@noarg ..... 3820
- \refstepcounter@optarg ..... 3821
- \relax ..... 52, 92
- \rightmark ..... 15
- \rmfamily ..... 58, 82, 83
- \roman ..... 31
- \scantokens ..... 161
- \scr@new@selectfont ..... 2219, 2234
- \scr@selectfont ..... 2231, 2235
- \scriptfont@name ..... 1181, 1188
- \section ..... 11, 17–19, 23–26, 28, 37, 96, 106, 114
- \section\* ..... 17
- \sectionmark ..... 15
- \sectionname ..... 18
- \sectionpage ..... 16, 115
- \selectfont ..... 66, 82, 87, 88, 92
- \setCJKfamilyfont ..... 32
- \setCJKmainfont ..... 7, 32
- \setcounter ..... 116
- \sf@size ..... 1180
- \sffamily ..... 58
- \size@update ..... 92, 2246, 2247, 2421, 2435
- \sldefault ..... 65
- \songti ..... 8
- \special ..... 7, 50, 84
- \split@name ..... 1141
- \SplitArgument ..... 98
- \ssf@size ..... 1182
- \strutbox ..... 129
- \subparagraph ..... 11, 23, 42
- \subsectionname ..... 18
- \subsectionpage ..... 16, 116
- \subst@correction ..... 1106, 1113
- \tablename ..... 14
- \textfont@name ..... 1179, 1187
- \textlangle ..... 52
- \textrangle ..... 52
- \tf@size ..... 1178
- \thechapter ..... 15, 28
- \thesection ..... 15, 19, 28, 114
- \thispagestyle ..... 99
- \titleformat ..... 110
- \titlespacing ..... 25, 110
- \today ..... 13
- \trans@languagepath ..... 4167, 4192
- \ttfamily ..... 58
- \ttl@a ..... 3488, 3496
- \ttl@chapterout ..... 3483
- \ttl@extract ..... 110
- \ttl@labelling ..... 3507
- \ttl@setifthe ..... 3553
- \ttl@setsubmark ..... 112, 3526, 3532
- \ttl@settopmark ..... 112, 3520, 3551
- \ttl@tocpart ..... 3485, 3487
- \ttlh@chang ..... 3449, 3459
- \ttlh@runin ..... 3448, 3458
- \tw@ ..... 414
- \uppercase ..... 52
- \use@mathgroup ..... 1192, 1223
- \usepackage ..... 36, 38
- \verb ..... 80
- \verb@eol@error ..... 1904
- \verbatim@font ..... 1890, 1894, 1905
- \vskip ..... 24
- \wrong@fontshape ..... 64
- \x@protect ..... 6300
- \xdef ..... 112
- \XeTeXglyphbounds ..... 152
- \xkanjiskip ..... 88, 89
- \yahei ..... 8, 146
- \youyuan ..... 8
- \z@ ..... 879, 882, 1910, 2891, 2953, 3008, 3073, 3098,  
3117, 3160, 3198, 3217, 3231, 3468, 3606, 3631, 3638,  
3647, 3656, 3664, 3681, 4382, 4390, 4401, 4421, 4429, 4440
- \zhdig ..... 94
- \zhdigits ..... 31
- \zhnum ..... 94
- \zhnumber ..... 31
- \zihao ..... 29
- \ziju ..... 31
- \zw ..... 87
- tex commands:
  - \tex\_afterassignment:D ..... 902
  - \tex\_chardef:D ..... 903
  - \tex\_def:D ..... 6376
  - \tex\_dimexpr:D ..... 105, 2472
  - \tex\_divide:D ..... 92, 2473, 2513
  - \tex\_edef:D ..... 6430

<code>\tex_endlinechar:D</code> .....	535	<code>\tl_gset_rescan:Nnn</code> .....	1581
<code>\tex_font:D</code> .....	482, 483, 5841, 5842	<code>\tl_head:N</code> .....	4340
<code>\tex_glueexpr:D</code> .....	2461	<code>\tl_head:n</code> .....	1625
<code>\tex_hangindent:D</code> .....	3264	<code>\tl_if_blank:nTF</code> .....	
<code>\tex_iftrue:D</code> .....	4119	.....	922, 1659, 1668, 1868, 1869, 1884, 2384, 2394
<code>\tex_ignorespaces:D</code> .....		<code>\tl_if_blank_p:n</code> .....	5875, 5876
.....	1356, 1522, 1527, 2331, 2497, 4209, 6186	<code>\tl_if_empty:nTF</code> .....	373, 1039, 2490,
<code>\tex_indent:D</code> .....	2780	.....	2674, 3801, 4514, 5062, 5249, 5268, 5282, 5290, 5300, 5305
<code>\tex_interlinepenalty:D</code> .....	2769	<code>\tl_if_empty:nTF</code> .....	109, 1944, 1958, 1971, 6347
<code>\tex_kanjiskip:D</code> .....	2152, 2168, 2182	<code>\tl_if_eq:NNTF</code> .....	2033, 2424
<code>\tex_noindent:D</code> .....	2770, 3262	<code>\tl_if_eq:NnTF</code> .....	4179
<code>\tex_numexpr:D</code> .....	923	<code>\tl_if_exist:nTF</code> ...	72, 761, 2008, 2011, 2012, 2013,
<code>\tex_pagegoal:D</code> .....	2788	.....	2020, 2021, 2022, 2589, 2620, 2631, 2634, 4113, 4202, 6199
<code>\tex_parindent:D</code> .....	2774, 2779, 3254	<code>\tl_if_in:NnTF</code> .....	6411
<code>\tex_parskip:D</code> .....	2796	<code>\tl_if_in:nnTF</code> .....	5032
<code>\tex_pdffontattr:D</code> .....	483	<code>\tl_if_novalue:nTF</code> .....	1483, 1506, 1862, 2738
<code>\tex_prevdepth:D</code> .....	2795	<code>\tl_map_inline:Nn</code> .....	1774
<code>\tex_setfontid:D</code> .....	1097	<code>\tl_map_inline:nn</code> .....	4337
<code>\tex_topskip:D</code> .....	2789	<code>\tl_new:N</code> .....	110, 112, 444, 489, 761, 905, 1041,
<code>\tex_Uchar:D</code> .....	911, 922, 923	.....	1042, 1133, 1250, 1300, 1327, 1344, 1345, 1366, 1716,
<code>\tex_undefined:D</code> .....	526	.....	1850, 2001, 2051, 2286, 2432, 2445, 2591, 2660, 2661,
<code>\tex_XeTeXcharglyph:D</code> .....	5848	.....	3339, 3340, 5010, 6093, 6193, 6194, 6199, 6387, 6388, 6389
<code>\tex_XeTeXglyphbounds:D</code> .....	5897	<code>\tl_put_left:Nn</code> .....	2237, 6216
<code>\tex_xkanjiskip:D</code> .....	2279, 2283, 2293, 2367	<code>\tl_put_right:Nn</code> .....	336, 716, 1184,
<b>text commands:</b>		.....	1857, 2225, 2226, 2230, 2690, 2728, 2730, 4340, 4341, 6210
<code>\l_text_case_exclude_arg_tl</code> .....	716	<code>\tl_remove_all:Nn</code> .....	1626
<code>\text_declare_expand_equivalent:Nn</code> ...	723, 724, 725	<code>\tl_replace_all:Nnn</code> .....	1672
<code>\text_declare_purify_equivalent:Nn</code>	749, 750, 751, 752	<code>\tl_rescan:nn</code> .....	163, 6429, 6438
<code>\text_expand:n</code> .....	54, 56	<code>\tl_set:Nn</code> .....	100,
<code>\text_purify:n</code> .....	56	.....	184, 518, 519, 520, 901, 1043, 1087, 1134, 1234, 1286,
<code>\text_uppercase:n</code> .....	54	.....	1349, 1361, 1505, 1669, 1671, 1712, 2053, 2215, 2271,
<code>\textasteriskcentered</code> .....	952	.....	2287, 2345, 2364, 2408, 2446, 2500, 2592, 2737, 2740,
<code>\the</code> .....	5490, 5493, 5494, 5513, 5517	.....	2744, 3445, 3464, 3483, 3488, 3496, 4468, 5025, 5864, 5866
<code>\thechapter</code> .....	3024, 3027, 3078, 3623, 3644, 3673, 3922	<code>\tl_set_eq:NN</code> .....	
<code>\theparagraph</code> .....	4024	.....	1048, 1049, 1103, 1179, 1181, 1350, 1375, 1812, 1824, 2433
<code>\thepart</code> .....	2876, 2895, 2917, 2937, 3861	<code>\tl_set_rescan:Nnn</code> .....	6385
<code>\thesection</code>	3606, 3614, 3631, 3638, 3647, 3656, 3664, 3681, 3941	<code>\tl_tail:N</code> .....	3450
<code>\thesubparagraph</code> .....	4038	<code>\tl_tail:n</code> .....	3460
<code>\thesubsection</code> .....	3609, 3641, 3659, 3973	<code>\tl_to_str:n</code> ..	108, 653, 660, 667, 6328, 6357, 6398, 6399
<code>\thesubsubsection</code> .....	4007	<code>\tl_trim_spaces:n</code> .....	1671
<code>\thispagestyle</code> .....	2804, 2856, 2994, 3005	<code>\tl_trim_spaces_apply:nN</code> .....	1841
<code>\tiny</code> .....	4410, 4449	<code>\tl_use:N</code> .....	425
<b>tl commands:</b>		<code>\l_tmpa_tl</code> .....	5025, 5027, 5028
<code>\c_novalue_tl</code> .....	6165	<code>tocdepth</code> .....	27, <u>3753</u>
<code>\tl_clear:N</code> .....	189, 352, 1844, 1848, 2120,	<code>\today</code> .....	2567, 2572, 2575, 2580
.....	2132, 2272, 2412, 2739, 2757, 2768, 4336, 5859, 5860, 6384	<code>today</code> .....	13, <u>2567</u>
<code>\tl_clear_new:N</code> .....	55, 329, 5879, 5880	<b>token commands:</b>	
<code>\tl_const:Nn</code> .....	120, 813, 1252,	<code>\c_parameter_token</code> .....	113, 3584, 3586, 3589
.....	1265, 1600, 2009, 2011, 2012, 2013, 2020, 2021, 2022,	<code>\token_if_eq_meaning:NNTF</code> .....	596, 610, 627
.....	2431, 2662, 2668, 4101, 4120, 4122, 5381, 5384, 5385, 6121	<code>\token_to_meaning:N</code> .....	6342
<code>\tl_gput_right:Nn</code> .....	423, 455, 762,	<code>\token_to_str:N</code> ...	67, 144, 929, 1161, 1169, 1178,
.....	2014, 2015, 2016, 2017, 2023, 2024, 2025, 6180, 6182, 6200	.....	1180, 1182, 1403, 1404, 1405, 1407, 1414, 1415, 1416,
<code>\tl_gset:Nn</code> .....		.....	1478, 1509, 2617, 2618, 4217, 5852, 6060, 6139, 6154, 6246
.....	218, 221, 224, 377, 547, 550, 1062, 1178, 1180,	<code>\toks</code> .....	5490, 5491, 5493, 5494, 5513, 5517
.....	1182, 1574, 2026, 2037, 4503, 4506, 4507, 4523, 4556, 5028	<code>\topsep</code> .....	4393, 4404, 4432, 4443
<code>\tl_gset_eq:NN</code> .....	1067, 2058	<code>\TrimSpaces</code> .....	2382

- `\ttdefault` ..... 2043  
`\ttfamily` ..... 2006  
`\twocolumn` ..... 2999  
`\typeout` ..... 3024, 3025, 3483
- U**
- ubuntu ..... 38  
 um commands:  
   `\um_input_math_symbol_table:` ..... 1216, 1220  
   `\um_sym:nnn` ..... 1219  
 um internal commands:  
   `\_um_input_math_symbol_table:` ..... 1228  
   `\_um_switchto_literal:` ..... 1225  
   `\_um_sym:nnn` ..... 1227  
`\undefined` ..... 5515  
`\UnicodeEncodingName` ..... 929  
`\unless` ..... 3161, 3199, 3218  
`\unskip` ..... 3234  
`\upshape` ..... 1934, 1937  
 use commands:  
   `\use:N` ..... 686,  
     692, 698, 704, 1093, 1094, 1178, 1180, 1182, 1270, 1319,  
     1825, 2804, 2816, 2822, 2831, 3248, 3311, 6412, 6432, 6446  
   `\use:n` .... 113, 492, 572, 649, 656, 663, 746, 747, 748,  
     1141, 1254, 1433, 1486, 1492, 1571, 1575, 1674, 2052,  
     3257, 3286, 3577, 4111, 4528, 6159, 6281, 6323, 6374, 6395  
   `\use:nn` ..... 6440  
   `\use_i:nn` ..... 97, 2810, 6227, 6232, 6252, 6257, 6267  
   `\use_i:nnn` ..... 239  
   `\use_ii:nn` ... 98, 2808, 2812, 6227, 6232, 6252, 6257, 6268  
   `\use_ii:nnn` ..... 114, 244  
   `\use_iii:nnn` ..... 249  
   `\use_none:n` 185, 190, 195, 500, 1120, 1335, 1574, 1709, 2553  
   `\use_none:nn` ..... 501  
   `\use_none_delimit_by_q_stop:w` .....  
     ..... 139, 142, 5038, 5058, 5078, 5120, 5135, 5195, 5315  
`\usebeamercolor` ..... 3695, 3714, 3733, 3904, 3964, 3998  
`\usebeamerfont` ..... 3694, 3705,  
   3713, 3724, 3732, 3743, 3903, 3906, 3963, 3966, 3997, 4000  
`\usefont` ..... 2007  
`\uselanguage` ..... 4158, 4163, 4183, 4188  
`\usepackage` ..... 6060  
 UTF8 ..... 6, 38, 215
- V**
- `\vadjust` ..... 1902, 1910  
`\verb` ..... 1900  
`\verse` ..... 4478  
`\vfil` ..... 2864, 2986  
`\vskip` ..... 2905, 2908,  
   2937, 2962, 2965, 2989, 3078, 3088, 3091, 3107, 3110,  
   3220, 3223, 3696, 3715, 3734, 3887, 3905, 3925, 3965, 3999  
`\vspace` ..... 2867, 3068, 3071, 3093, 3096
- W**
- windows ..... 38  
 windowsnew ..... 38  
 windowsold ..... 38  
 winfonts ..... 36  
`\write` ..... 5530
- X**
- `\x` ..... 407, 413, 417, 5492, 5495,  
   5496, 5497, 5498, 5499, 5500, 5501, 5502, 5503, 5504,  
   5505, 5506, 5507, 5508, 5509, 5510, 5511, 5518, 5528, 5532  
 x commands:  
   `\x:` ..... 412  
`\xdef` ..... 5531  
`\xeCJKsetup` ..... 819, 2128, 2326, 2329, 2332, 2347, 2359
- Y**
- `\yahei` ..... 5452  
`\youyuan` ..... 5452
- Z**
- `\zhdig` ..... 2559  
`\zhdigits` ..... 31, 2566  
`zhmap` ..... 7, 36, 234  
`\zhnum` ..... 2558  
 zhnum commands:  
   `\zhnum_counter:n` ..... 2550  
`\zhnumber` ..... 31, 2564  
`\zhnumsetup` ..... 2576, 2581  
`\zhtoday` ..... 2575, 2580  
`\zihao` ..... 29, 4208, 4217  
`zihao` ..... 10, 164  
`\ziju` ..... 31, 2496  
`\zw` ..... 880, 2149, 2200, 2288