

CT_EX 宏包说明

ctex.org

2015/03/26 v2.0*

简介

CT_EX 宏包是面向中文排版的通用 L^AT_EX 排版框架,为中文 L^AT_EX 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_EX 宏包支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX 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 或中文文档类 ctexart、ctexrep、ctexbook 实现。

目录

1 介绍	2 5 开发人员	23
2 简明教程	2 6 参考文献	24
2.1 使用 CT _E X 文档类	2	
2.2 使用其他文档类	3	
3 参考手册	7 代码实现	25
3.1 依赖与安装	7.1 内部函数与变量	25
3.2 CT _E X 宏包与文档类	7.2 宏包选项	29
3.3 宏包选项	7.3 用户设置接口	33
3.4 格式控制命令 \ctexset	7.4 引擎支持	33
3.5 中文文字支持	7.5 中文字号	62
3.6 文档汉化与中文版式	7.6 中文数字与日期	68
3.7 关于 LuaL ^A T _E X 下的中文支持 方式	7.7 其它中文标题定义	69
3.8 杂项	7.8 中文化的标题结构	69
4 对旧版本的兼容性	7.9 其它功能	81
4.1 CT _E X 0.8a 及以前的版本	7.10 载入中文字体	83
4.2 CT _E X 0.9–CT _E X 1.0d	7.11 宏包配置文件	84
4.3 CT _E X 1.02c 以后的 SVN 开发版	7.12 字体定义文件	84
	8 版本历史	99
	9 代码索引	99

*ctex-kit rev. b1a1ac3.

1 介绍

这个宏包的部分原始代码来自于由王磊编写 `cjkbook` 文档类, 还有一小部分原始代码来自于吴凌云编写的 `GB.cap` 文件。原来的这些工作都是零零碎碎编写的, 没有认真、系统的设计, 也没有用户文档, 非常不利于维护和改进。2003 年, 吴凌云用 `doc` 和 `DocStrip` 工具重新编写了整个文档, 并增加了许多新的功能。2007 年, osean 和王越在 $\text{CT}_{\text{E}}\text{X}$ 宏包基础上增加了对 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¹, 对 $\text{CT}_{\text{E}}\text{X}$ 宏包及相关宏包和脚本进行了整合, 并加入了对 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的支持。在开发新版本时, 考虑到合作开发和调试的方便, 我们不再使用 `doc` 和 `DocStrip` 工具, 改为直接编写宏包文件。2014 年 3 月, 为了适应 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的最新发展, 特别是 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}3$ 的逐渐成熟, 李清用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}3$ 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。2015 年 3 月, `ctex-kit` 项目迁移至 [GitHub](https://github.com/CTeX-org/ctex-kit)²。

最初 Knuth 设计开发 $\text{T}_{\text{E}}\text{X}$ 的时候没有考虑到支持多国语言, 特别是多字节的中日韩语言。这使得 $\text{T}_{\text{E}}\text{X}$ 以至后来的 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 仍然要面对许多困难。最常见的就是中文化的标题。由于中文习惯和西方语言的不同, 使得很难直接使用原有的标题结构来表示中文标题, 因此需要对标准 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包做较大的修改。此外, 还有诸如日期格式、首行缩进、中文字号的对应关系等等细节问题。 $\text{CT}_{\text{E}}\text{X}$ 宏包正是为了解决这些 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 文档中文化问题而产生的。另一方面, 随着 $\text{T}_{\text{E}}\text{X}$ 引擎和 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包的不断开发, 在 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 中支持中文字符的方式也由早期的 CCT 这类单一的专用系统发展为 `pdfT_{\text{E}}\text{X}` 引擎下的 CJK、zhmCJK 宏包、 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 引擎下的 `xeCJK` 宏包、 $\text{LuaT}_{\text{E}}\text{X}$ 引擎下的 `luatexja` 宏包等多种方式, 各种方式下都有不同的适用范围和用法细节, 不同操作系统和语言环境也带来不少细节差异, 亟需一个统一的界面来访问不同的中文处理方式, 使同一份文档能在不同的环境下交换使用。 $\text{CT}_{\text{E}}\text{X}$ 宏包也在统一中文处理界面上做出了努力。

实现 $\text{CT}_{\text{E}}\text{X}$ 宏包中间很多地方用到了在 bbs.ctex.org 论坛上的讨论结果, 在此对参与讨论的朋友们表示感谢。

2 简明教程

2.1 使用 $\text{CT}_{\text{E}}\text{X}$ 文档类

$\text{CT}_{\text{E}}\text{X}$ 宏包的使用十分简单。对于使用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 标准文档类 `article`、`report`、`book` 的用户, 希望通过 $\text{CT}_{\text{E}}\text{X}$ 宏包实现中文及中文版式的支持, 只需用 `ctexart`、`ctexrep` 或者 `ctexbook` 替换原有的文档类即可。

2.1.1 使用 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 或 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译

使用 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 或 $\text{LuaL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 编译时, 必须将涉及到的所有源文件使用 UTF-8 编码保存。

例 1

```
\documentclass{ctexart}
\begin{document}
中文文档类测试, 保存为 UTF-8 编码, 使用 XeLaTeX 或 LuaLaTeX 编译。
\end{document}
```

2.1.2 使用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 或 `pdfL_{\text{A}}\text{T}_{\text{E}}\text{X}` 编译

在默认情况下, 使用 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 或 `pdfL_{\text{A}}\text{T}_{\text{E}}\text{X}` 编译时, 需将涉及到的涉及到的所有源文件使用 GBK 编码保存。

¹<http://code.google.com/p/ctex-kit/>

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

例 2

```
\documentclass{ctexart}
\begin{document}
中文档类测试，保存为 GBK 编码，使用 LaTeX 或 pdfLaTeX 编译。
\end{document}
```

为 C_TE_X 宏包或文档类添加 UTF8 选项，则使用 L^AT_EX 或 pdfL^AT_EX 编译时，也可以让 C_TE_X 宏包或文档类在 UTF-8 模式下工作。

例 3

```
\documentclass[UTF8]{ctexart}
\begin{document}
中文档类测试，保存为 UTF-8 编码，使用 LaTeX 或 pdfLaTeX 编译。
\end{document}
```

2.2 使用其他文档类

对于 article、report、book 之外的文档类，则可以载入 ctex 包来使用 C_TE_X 宏包。

例如，编写中文的 beamer 演示，就可以使用：

例 4

```
\documentclass{beamer}
\usepackage{ctex}
\begin{document}
\begin{frame}{中文演示文档}
\begin{itemize}
\item 保存为 UTF-8 编码，使用 XeLaTeX 或 LuaLaTeX 编译。
\item 保存为 GBK 编码，使用 LaTeX 或 pdfLaTeX 编译。
\end{itemize}
\end{frame}
\end{document}
```

这里，ctex 包也可以加上 UTF8 选项指定文档编码。

如果使用的文档类本身还是基于 L^AT_EX 标准文档类生成的，则还可以给 ctex 包加上 heading 选项，设置其章节标题的格式。

例 5

```
\documentclass{ltxdoc}
\usepackage[UTF8,heading]{ctex}
\begin{document}
\section{简介}
中文化的 \LaTeX{} 手册。
\end{document}
```

3 参考手册

C_TE_X 宏包会根据用户使用的编译方式³在底层选择不同的中文支持方式（见表 1）。

(pdf)L ^A T _E X	X _Ǝ L ^A T _E X	LuaL ^A T _E X
CJK	xeCJK	luatexja

表 1: C_TE_X 宏包的中文支持方式

C_TE_X 宏包在用户使用 X_ƎL^AT_EX 及 LuaL^AT_EX 编译时，使用（且仅能使用）UTF-8 编码；而因为历史原因，在用户使用 L^AT_EX 及 pdfL^AT_EX 编译时默认使用 GBK 编码。除非有特殊的需求，我们推荐用户使用 UTF-8 编码，并使用 X_ƎL^AT_EX 或 LuaL^AT_EX 编译。

³L^AT_EX、pdfL^AT_EX、X_ƎL^AT_EX 以及 LuaL^AT_EX。

3.1 依赖与安装

CT_EX 是一个 L^AT_EX 宏包, 只有一个源文件 `ctex.dtx`。它依赖下列宏包:

- `expl3`、`xparse` 和 `l3keys2e` 宏包。它们属于 `l3kernel` 和 `l3packages` 集合, 并依赖
 - `etex` 宏包。
 - `ifpdf` 宏包, 属于 `oberdiek` 集合。
 - `etoolbox` 宏包。
 - `everyysel` 宏包, 属于 `ms` 集合。
 - `zhnumber` 宏包。
- ⇒ 以上是各种编译方式都必需的依赖项。
- `CJK` 集合, 它的下划线功能依赖 `ulem` 宏包。
 - `CJKpunct` 宏包。
 - `zhmetrics` 宏包。
 - `zhmCJK` 宏包。
- ⇒ 以上是使用 `pdfLATEX` 或 `LATEX + DVIPDFMx` 的编译方式所需要的依赖项, 其中 `zhmCJK` 是可选的。
- `xeCJK` 宏包, 它还依赖
 - `fontspec` 宏包, 它还依赖
 - * `euenc` 宏包。
 - * `xunicode` 宏包, 它还依赖 `tipa` 宏包。
 - `everypage` 宏包。
- ⇒ 以上是使用 `XYLATEX` 编译时的依赖项。
- `luatexja` 集合, 它还依赖
 - `luaotfload` 宏包, 它还依赖 `luatexbase` 宏包。
 - `xkeyval` 宏包。
- ⇒ 以上是使用 `LuaLATEX` 编译时的依赖项。

CT_EX 正式发布的版本和依赖的各个宏包或者集合都会被发行版 T_EX Live 或者 MiK_TE_X 收录, 可以直接用它们提供的宏包管理器安装或者更新, 不需要自行下载安装。

由于种种原因, `zhmCJK` 目前还没有收入 T_EX Live 和 MiK_TE_X 发行版。因此, 如果希望使用 `zhmCJK` 作为底层中文支持方式, 还需要自行安装该宏包。`zhmCJK` 的安装较为复杂, 可以从 CTAN 下载 `zhmCJK` 的 **TDS 安装包**, 按目录结构将文件复制到 T_EX 发行版的本地 TDS 根目录, 然后刷新文件名数据库。关于安装的详细内容, 可以参照 **宏包手册** 中第 3 节的指导。

3.2 CT_EX 宏包与文档类

CT_EX 宏包的组成见表 2。

类别	文件	说明
文档类	<code>ctexart.cls</code>	标准文档类 <code>article</code> 的中文化版本, 一般适用于短篇幅的文章
	<code>ctexrep.cls</code>	标准文档类 <code>report</code> 的中文化版本, 一般适用于中篇幅的报告
	<code>ctexbook.cls</code>	标准文档类 <code>book</code> 的中文化版本, 一般适用于长篇幅的书籍
格式	<code>ctex.sty</code>	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

表 2: CT_EX 宏包的组成

除此以外,CT_EX 宏包定义和调整中文字号的功能被提取到 `ctexsize.sty` 当中,可以在 CT_EX 之外调用 `ctexsize` 宏包使用该功能。它可以使用随后介绍的宏包选项 `cs4size`、`c5size` 和命令 `\zihao`。

我们推荐用户直接使用 CT_EX 宏包提供的三个中文文档类,而不是在使用 L^AT_EX 标准文档类的基础上,再使用 `ctex` 宏包。`ctex` 宏包被设计为适配非标准文档类(例如 `beamer` 文档类);而它的 `heading` 选项则被设计为适配衍生自标准文档类的文档类(例如 `ltxdoc` 文档类)。

3.3 宏包选项

宏包的选项用于改变一些默认的设置。CT_EX 宏包和文档类的默认设置已经对中文行文和排版习惯做了尽可能的配置,因此除非必要,普通用户应尽量避免对默认设置的修改。如果你觉得某些默认设置不合适,或者需要增减选项,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。

CT_EX 宏包有部分选项以 `(key)=(value)` 的形式提供,剩余部分是传统的选项。你应该在调用宏包的时候直接设置这些选项。在下面的说明中使用**粗体**来表示 CT_EX 的默认设置。

CT_EX 文档类基于 L^AT_EX 的标准文档类。因此,它们除了可以使用下面介绍的选项之外,还能够使用标准文档类的选项。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。CT_EX 会将这些选项传给标准文档类⁴。

3.3.1 默认字号选项

<code>cs4size</code>	分别使用小四号字或五号字作为默认字体大小。CT _E X 默认设置 <code>c5size</code> 。这两个选项在 <code>ctexsize</code> 宏包中也有定义。
<code>c5size</code>	
<code>10pt</code>	可以使用标准文档类的同类选项(<code>10pt</code> 、 <code>11pt</code> 和 <code>12pt</code>)抑制中文字号设置。
<code>11pt</code>	
<code>12pt</code>	

`cs4size` 和 `c5size` 选项还会将标准文档类中的字体大小命令调整为中文字号(见表 3)。

字体命令	c5size		cs4size		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 3: 标准字体命令与字号的对应

⁴事实上,L^AT_EX 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。

3.3.2 中文标题选项

heading	<code>heading = <true false></code> 是否启用 <code>ctex.sty</code> 宏包的标题格式与页眉设置功能。参见 3.6.4 节、3.6.5 节。 CT _E X 宏包提供的三个文档类总是启用中文标题设置功能。 <code>heading</code> 选项只在 <code>ctex.sty</code> 下有意义。如果在 <code>ctex.sty</code> 下启用该选项, 将会检查使用的是否为 L ^A T _E X 标准文档类。若是, 则该选项将会使得 <code>ctex.sty</code> 宏包的行为和 CT _E X 宏包提供的三个中文文档类完全一致; 否则, 会根据 <code>\chapter</code> 是否有定义来使用 <code>ctexbook</code> 或者 <code>ctexart</code> 的标题设置。
sub3section	将 <code>\paragraph</code> 标题改为类似 <code>section</code> 的格式(感觉类似 <code>\sub³section</code>), 即标题单独占一行。此时 <code>\subparagraph</code> 命令产生的标题会具有原本 <code>\paragraph</code> 的格式。 具体格式可参考 3.6.4 节中 <code>afterskip</code> 等选项。 此选项通常需要配合将计数器 <code>secnumdepth</code> 的值为设置为 4。
sub4section	将 <code>\paragraph</code> 和 <code>\subparagraph</code> 都改为 <code>section</code> 类的格式(感觉类似 <code>\sub⁴section</code>), 即标题单独占一行。 具体格式可参考 3.6.4 节中 <code>afterskip</code> 等选项。 此选项通常需要配合将计数器 <code>secnumdepth</code> 的值为设置为 4 或 5。 这两个选项在文档类和启用 <code>heading</code> 的 <code>ctex.sty</code> 下才有意义。

3.3.3 中文编码选项

下面的选项用于选择 CT_EX 宏包内部的编码, 以匹配用户编写文档使用的编码。

GBK UTF8	使用 (pdf)L ^A T _E X 编译时, CT _E X 的默认编码是 GBK。此时可以使用 UTF8 选项将默认编码改为 UTF-8。而使用 X _Y L ^A T _E X 或 LuaL ^A T _E X 编译时, CT _E X 内部总是使用 UTF-8 编码, 所以不必指明这些选项。
-------------	---

3.3.4 中文字库选项

默认情况下, CT_EX 宏包会根据用户使用的操作系统⁵自动选择设置的字体(见表 4)。
需要注意的是, 在 pdf_ET_EX 引擎下, CT_EX 的默认字体设置只支持 pdfL^AT_EX 或者 L^AT_EX + DVIPDFM_x 的编译方式。如果需要使用 Dvips, 就需要将下面介绍的 `fontset` 选项设置为 `none`, 然后按照传统方式⁶在本地安装好 CJK 字体。
Mac OS X 系统下的华文字体在 (pdf)L^AT_EX 下不可用, 只支持 X_YL^AT_EX 或者 LuaL^AT_EX 的编译方式。Fandol 在 pdfL^AT_EX 下不可用。

Mac OS X	Windows	其他
华文字体	中易字体	Fandol 字体

表 4: CT_EX 宏包在默认选项下的字体

通常, 由 CT_EX 宏包进行的自动配置已经足够使用, 无需用户手工干预; 但是 CT_EX 仍然提供了一系列选项, 供在 CT_EX 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。

zhmap	<code>zhmap = <true false zhmCJK></code> 是否使用 <code>zhmetrics</code> 宏包提供的字体映射机制, 将 CJK 中文字库通过 <code>\special</code> 命令映射到对应的 <code>.ttf</code> 字体文件。
-------	--

⁵CT_EX 宏包现在能够识别 Mac OS X 系统以及 Windows 系统。
⁶可以使用 `zhmetrics` 宏包提供的脚本 `CTeXFonts.lua`。

zhmCJK 还将载入 zhmCJK 宏包。它基于 zhmetrics 机制和 CJK 宏包,提供了与 xeCJK 类似的用户界面,可以很方便的完成字体安装设置工作。

如果需要使用自定义的字体映射文件,或者希望使用 Type1 字库,请禁用本选项。此时,将会使用传统的字体映射机制,调用系统中的字体映射文件。

本选项只在使用 (pdf)L^AT_EX 编译时有意义。

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

如果没有指定 fontset 的值,CT_EX 宏包会根据用户使用的操作系统配置相应的字体。目前 CT_EX 能够识别 Mac OS X 和 Windows 系统。

CT_EX 预定义了以下六种中文字库。它们之中除了 windows,在使用 (pdf)L^AT_EX 编译但没有启用 zhmap 选项都没有定义,都会返回错误。此时对于 windows 来说,假设已经按照传统方式在本地安装好了 CJK 字体。

adobe	使用 Adobe 公司的四款中文字体,不支持 pdfL ^A T _E X。
fandol	使用 Fandol 中文字体 ⁷ ,不支持 pdfL ^A T _E X。
founder	使用方正公司的中文字体。
mac	使用 Mac OS X 系统下的华文字体,只能用于 X _Y L ^A T _E X 和 LuaL ^A T _E X。
ubuntu	使用 Ubuntu 系统下的文泉驿和文鼎字体。
windows	使用简体中文 Windows 系统下的中易字体。

如果不想使用 CT_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。CT _E X 宏包将载入名为 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 CT _E X 宏包 fontset 目录下的字体配置文件。

3.3.5 排版风格选项

cap cap = `<true|false>`

是否对标题中英文文字进行汉化(如“图”、“表”、“目录”、“参考文献”等,见 3.6.3 节)并设置章节标题的中文化格式(见 3.6.4 节)。

cap 选项控制的标题汉化总是生效的,而对章节标题的中文化格式设定仅在 heading 选项(3.3.2 节)打开时生效。

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:句末点号用全角,其他半角;
CCT	CCT 格式: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)。

⁷由马起园、苏杰、黄晨成等人开发的开源中文字体,参见:<https://github.com/clerkma/fandol-fonts>。

<hr/> <code>space</code> <hr/>	<code>space = <true false auto></code> 用于控制是否保留汉字后面的空格。
<code>true</code>	保留汉字后面的空格, 类似英文的习惯。这时用户需要自己处理由换行产生的空格(在行尾加上 % 可以避免), 否则排版结果可能不符合中文习惯。
<code>false</code>	总是忽略掉汉字后面的空格。只在使用 (pdf)LaTeX 编译时有意义。不建议使用该选项。
<code>auto</code>	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。 目前该选项对 xeCJK 与 CJK 宏包有效, 相当于 \CJKspace、\CJKnospace 等宏的作用。 使用 LuaLaTeX 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。
<hr/> <code>linespread</code> <hr/>	<code>linespread = <数值></code> 接受一个浮点数值, 用于调整行距。初始值为 1.3, 即设置基线距离(\baselineskip)为 $1.3 \times 1.2 = 1.56$ 倍字体大小。
<hr/> <code>autoindent</code> <hr/>	<code>autoindent = <true false></code> 在字体大小发生变化时, 是否自动调整段首缩进(\parindent)的大小。

3.3.6 宏包兼容选项

<hr/> <code>fancyhdr</code> <hr/>	<code>fancyhdr = <true false></code> 保持和 fancyhdr 宏包的兼容性。启用该选项将自动调用 fancyhdr 宏包, 并设置默认页面格式(page style)为 fancy。参见 3.6.5 节。
<hr/> <code>fntef</code> <hr/>	<code>fntef = <true false></code> 启用该选项将自动调用 CJKfntef 或 xeCJKfntef 宏包, 同时禁止相关宏包对 \emph 的影响, 禁用彩色设置。
<hr/> <code>hyperref</code> <hr/>	<code>hyperref = <true false></code> 自动判断 hyperref 宏包的正确参数以避免产生乱码。如果在导言区用户没有自己调用 hyperref, 则该选项将使得 hyperref 宏包在导言区末尾被自动调用; 如果需要对 hyperref 宏包做进一步的设置, 则用户可以自己在 CT _E X 宏包后使用 \hypersetup 命令进行适当设置。

3.4 格式控制命令 \ctexset

CT_EX 宏包为用户提供了一个通用的文档框架, 使得用户可以根据需求, 自由地在底层不同的中文支持方式之间切换。为此, 我们定义了一些命令, 使得用户可以用统一的方式对文档进行控制。

<hr/> <code>\ctexset</code> <hr/>	<code>\ctexset {<键值列表>}</code> 是 CT _E X 宏包的通用控制命令, 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。如非特别说明, 后文介绍的所有选项都使用 \ctexset 命令进行设置。 \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(3.6.3 节)就可以使用:
-----------------------------------	--

例 6

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


`\ctexset` 采用 \LaTeX 3 风格的键值设置, 因此支持不同类型的选项与层次化的选项设置, 相关示例见 3.6.4 节。

3.5 中文文字支持

\CTEX 宏包为用户提供了一个通用的文档框架, 使得用户可以根据需求, 自由地在底层不同的中文支持方式之间切换。因此, 一份文档可以几乎不做修改地在不同的操作系统设置、不同的 \TeX 引擎下编译, 正常地输出中文文档, 而无须考虑具体的底层细节。

在不同的操作系统设置与 \TeX 引擎下, \CTEX 宏包需要在底层使用不同的中文支持宏包, 它们的字体、字号和文字修饰的命令可能各不相同。为此, 我们定制了一些命令, 使得用户可以用统一的方式对文档中的文字格式进行控制。

3.5.1 中文字体命令

\CTEX 宏包预定义的中文字库基本上都定义了以下四种常用的中文字体命令。

`\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}`。

需要注意的是 `\fangsong` 在 ubuntu 字库下没有定义。而在 windows 和 founder 字库下, 还有 `\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 字库下, 还有 `\yahei`。

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

3.5.2 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 5 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。

`\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。

`\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。

〈字号〉	大小 (bp)	意义
0	42	初号
-0	36	小初号
1	26	一号
-1	24	小一号
2	22	二号
-2	18	小二号
3	16	三号
-3	15	小三号
4	14	四号
-4	12	小四号
5	10.5	五号
-5	9	小五号
6	7.5	六号
-6	6.5	小六号
7	5.5	七号
8	5	八号

表 5: 中文字号

`space` `space = {true|false|auto}`

与宏包选项 `space` 一样,用于在正文中控制是否保留汉字后面的空格。

例如,使用

例 7

```
\ctexset{space=true}
汉字 分词 技术
```

将得到“汉字 分词 技术”。

`autoindent` `autoindent = {true|false}`

与宏包选项 `autoindent` 一样,在正文中控制在字体大小发生变化时,是否自动调整段首缩进 (`\parindent`) 的大小。

如果设置 `autoindent` 为假,则方便设置不受字号变化的特殊段落缩进。如设置每段缩进 4 个汉字宽:

例 8

```
\ctexset{autoindent=false}
\setlength\parindent{4\ccwd}
```

`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量。此时汉字间的弹性间距则固定为 `\baselineskip` 的 0.08 倍。

3.6 文档汉化与中文版式

3.6.1 中文数字转换

CT_EX 宏包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)}`

`\chinese` 命令与 `\roman` 等命令的用法类似,作用在一个 L^AT_EX 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_EX 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 $\langle number \rangle$ 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。

<code>\CTEXdigits</code>	<code>\CTEXdigits \langle macro \rangle \{\langle number \rangle\}</code>
--------------------------	---

`\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。

3.6.2 日期汉化

`CTEX` 宏包对显示当前日期的 `\today` 命令进行了汉化,使之以中文的方式显示今天的日期。如编译本文档的日期就是“2015 年 3 月 26 日”。

<code>today</code>	<code>today = \langle small big old \rangle</code>
--------------------	--

该选项用来控制 `\today` 命令的输出格式:

`small` 效果为“2015 年 3 月 26 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇一五年三月二十六日”。使用全汉字的日期格式。

`old` 效果为“March 26, 2015”。使用文档原来的(英文)日期格式。

设置日期格式使用 `\ctexset` 命令完成,例如设置全汉字的日期格式:

例 9

`\ctexset{today=big}`

`CTEX` 宏包的中文日期功能实际上也是调用 `zhnumber` 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 `zhnumber` 宏包的文档。

3.6.3 文档标题汉化

这里主要介绍由宏包 `cap` 选项(3.3.5 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 6。下面的选项(如 `contentsname`)主要用来重新定义与选项同名的宏(如 `\contentsname`)的定义。

<code>contentsname</code>	<code>contentsname = \langle 名字 \rangle</code>
---------------------------	--

设置目录标题名 `\contentsname`。中文默认为“目录”。

<code>listfigurename</code>	<code>listfigurename = \langle 名字 \rangle</code>
-----------------------------	--

设置插图目录标题名 `\listfigurename`。中文默认为“插图”。

<code>listtablename</code>	<code>listtablename = \langle 名字 \rangle</code>
----------------------------	---

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

<code>figurename</code>	<code>figurename = \langle 名字 \rangle</code>
-------------------------	--

设置图片环境标题名 `\figurename`。中文默认为“图”。

<code>tablename</code>	<code>tablename = \langle 名字 \rangle</code>
------------------------	---

设置表格环境标题名 `\tablename`。中文默认为“表”。

<code>abstractname</code>	<code>abstractname = \langle 名字 \rangle</code>
---------------------------	--

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。

<code>indexname</code>	<code>indexname = \langle 名字 \rangle</code>
------------------------	---

设置索引标题名 `\indexname`。中文默认为“索引”。

<hr/> appendixname <hr/>	appendixname = {名字} 设置附录标题名\appendixname。中文默认为“附录”。
<hr/> bibname <hr/>	bibname = {名字} 设置参考文献标题名\refname(对 article)或\bibname(对 report 和 book)。中文默认为“参考文献”。

3.6.4 章节标题格式定义

CT_EX 宏包对 L^AT_EX 的标准文档类(article、report 和 book)进行了扩充。当以 heading 选项调用 CT_EX 宏包时(3.3.2 节),则会启用章节标题的格式设置功能。本节就来介绍有关章节标题的格式选项,所有选项使用 \ctexset 命令设置。

章节标题的格式选项是分层设置的。顶层的选项是章节标题名称,次一级的选项是章节标题的格式。章节标题名包括 part, chapter, section, subsection, subsubsection, paragraph, subparagraph;而可用的格式包括 name, number, format, nameformat, numberformat, aftername, titleformat, beforeskip, afterskip, indent, pagestyle 等。但注意,对 article 及其衍生的 ctexart 等文档类,没有 chapter 级别的标题。

多级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。

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

name = {<前名字>,<后名字>}
name = {<前名字>}

设置章节的名字。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 10

```
\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。

标题名	cap=true 的默认值	cap=false 的默认值	注
part	{第,部分}	{\partname\space}	原 \partname 为 Part
chapter	{第,章}	{\chaptername\space}	原 \chaptername 为 Chapter
section	{}	{}	
subsection	{}	{}	
subsubsection	{}	{}	
paragraph	{}	{}	
subparagraph	{}	{}	

表 6: name 选项的默认设置

```

part/number
chapter/number
section/number
subsection/number
subsubsection/number
paragraph/number
subparagraph/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 11

```

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

```

number 选项的定义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^AT_EX 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。)

标题名	cap=true 的默认值	cap=false 的默认值	原 \the⟨标题⟩ 等价定义
part	<code>\chinese{part}</code>	<code>\thepart</code>	<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	同右	<code>\thesection</code>	<code>\arabic{section}</code>
subsection	同右	<code>\thesubsection</code>	<code>\thesection.\arabic{subsection}</code>
subsubsection	同右	<code>\thesubsubsubsection</code>	<code>\thesubsection.\arabic{subsubsection}</code>
paragraph	同右	<code>\theparagraph</code>	<code>\thesubsubsubsection.\arabic{paragraph}</code>
subparagraph	同右	<code>\thesubparagraph</code>	<code>\theparagraph.\arabic{subparagraph}</code>

表 7: number 选项的默认设置

```

\CTEXthepart
\CTEXthechapter
\CTEXthesection
\CTEXthesubsection
\CTEXthesubsubsubsection
\CTEXtheparagraph
\CTEXthesubparagraph

```

以 `\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 cap 选项下的默认章编号输出格式就是 `\CTEXthechapter`，形如“第一章”。

这组宏在 C_TE_X 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。

使用 `\ctexset` 设置多级选项时，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 section 一级标题的 name 与 number 选项：

例 12

```

\ctexset{
  section = {
    name = {第, 节},
    number = \chinese{section}
  }
}

```


part/format

chapter/format

section/format

subsection/format

subsubsection/format

paragraph/format

subparagraph/format

.../format+

format = {<格式命令>}

format+= {<格式命令>}

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+ 选项则用于在已有格式之后追加新的格式命令。

例如,设置章格式为无衬线字体左对齐,为节格式增加无衬线字体设置:

例 13

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

标题名	cap=true 的默认值	cap=false 的默认值
part (article)	\centering	\raggedright
part	\centering	\centering
chapter	\centering	\raggedright
section	\Large\bfseries\centering	\Large\bfseries
subsection	同右	\large\bfseries
subsubsection	同右	\normalsize\bfseries
paragraph	同右	\normalsize\bfseries
subparagraph	同右	\normalsize\bfseries

表 8: format 选项的默认设置

part/nameformat

chapter/nameformat

section/nameformat

subsection/nameformat

subsubsection/nameformat

paragraph/nameformat

subparagraph/nameformat

.../nameformat+

nameformat = {<格式命令>}

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 titleformat 选项。

nameformat+ 用于在已有的章节名字格式后附加内容。

标题名	cap=true 的默认值	cap=false 的默认值
part (article)	同右	\Large\bfseries
part	同右	\huge\bfseries
chapter	同右	\huge\bfseries
section	同右	{}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

表 9: nameformat 选项的默认设置

part/numberformat
chapter/numberformat
section/numberformat
subsection/numberformat
subsubsection/numberformat
paragraph/numberformat
subparagraph/numberformat
.../numberformat+

numberformat = {<格式命令>}
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，当你需要编号的格式和前后的章节名字不一样时可以使用。

numberformat+ 选项用于在原有编号格式后面附加格式命令。

例如，我们可以使用 numberformat 特别强调章标题中的数字：

例 14

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

上面的代码在 cap 选项下可以做出类似这样的章标题效果：

第 4 章

part/aftername
chapter/aftername
section/aftername
subsection/aftername
subsubsection/aftername
paragraph/aftername
subparagraph/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。

标题名	cap=true 的默认值	cap=false 的默认值
part (article)	\quad	\par\nobreak
part	同右	\par\vskip 20pt
chapter	\quad	\par\vskip 20pt
section	同右	\quad
subsection	同右	\quad
subsubsection	同右	\quad
paragraph	同右	\quad
subparagraph	同右	\quad

表 10: aftername 选项的默认设置

part/titleformat
chapter/titleformat
section/titleformat
subsection/titleformat
subsubsection/titleformat
paragraph/titleformat
subparagraph/titleformat
.../titleformat+

titleformat = {<格式命令>}
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+ 选项用于在原有标题格式后面附加格式命令。

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

beforeskip = {<弹性间距>}

beforeskip 选项用于设置章节标题前的垂直间距及章节标题后首段的缩进。

<弹性间距> 的绝对值被用于设置标题间的垂直间距，而<弹性间距>的正负号用于设置标题后第一段的首行缩进。当参数是负值时，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；参数是正值时，则保留首行缩进。

使用 sub3section 或 sub4section 宏包选项（见 3.3.2 节）后，\paragraph 与 \subparagraph 这两级标题会改为排在不同段，会影响 beforeskip 选项的默认值。

标题名	cap=true 的默认值	cap=false 的默认值
part (article)	\Large\bfseries	\huge\bfseries
part	\huge\bfseries	\Huge\bfseries
chapter	\huge\bfseries	\Huge\bfseries
section	同右	{}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

表 11: titleformat 选项的默认设置

标题名	cap=true 的默认值	cap=false 的默认值
part (article)	4ex	-4ex
part	无效	无效
chapter	50pt	-50pt
section	3.5ex plus 1ex minus .2ex	-3.5ex plus -1ex minus -.2ex
subsection	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
subsubsection	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
paragraph	同右	3.25ex plus 1ex minus .2ex
(sub3section)	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex
(sub4section)	同上	同上
subparagraph	同右	3.25ex plus 1ex minus .2ex
(sub4section)	3.25ex plus 1ex minus .2ex	-3.25ex plus -1ex minus -.2ex

表 12: beforeskip 选项的默认设置

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

afterskip = {⟨弹性间距⟩}

afterskip 选项控制章节标题与后面下方之间的距离。⟨弹性间距⟩的正负号确定标题与后面正文是否排在同一段。如果是正值则正文另起一段,⟨弹性间距⟩给出垂直间距;如果是负值则章节标题与正文第一段排在同一段,⟨弹性间距⟩的绝对值给出水平间距。

默认情况下, \paragraph、\subparagraph 两级标题是与后面正文排在同一段的, afterskip 选项取负数;但使用 sub3section 或 sub4section 宏包选项(见 3.3.2 节)后,则这两级标题会改为排在不同段。

afterskip 选项的默认值,在 cap 选项的不同取值下相同。

标题名	默认值
part (article)	3ex
part	无效
chapter	40pt
section	2.3ex plus .2ex
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

表 13: afterskip 选项的默认设置

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

indent = {⟨缩进间距⟩}

indent 选项用于设置章节标题本身的缩进。

indent 选项的默认值, 在 cap 选项的不同取值下相同。

标题名	默认值
part (article)	0pt
part	无效
chapter	0pt
section	0pt
subsection	0pt
subsubsection	0pt
paragraph	0pt
subparagraph	\parindent
(sub3section)	0pt
(sub4section)	0pt

表 14: indent 选项的默认设置

part/pagestyle
chapter/pagestyle

pagestyle = {⟨页面格式⟩}

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)。

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

表 15: pagestyle 选项的默认设置

appendix/name

name = {⟨前名字⟩, ⟨后名字⟩}

name = {⟨前名字⟩}

设置附录章 (对 book 与 report) 或附录节 (对 article) 的名字。用法与普通章节 name 选项类似。

注意该选项与 appendixname 选项 (3.6.3 节) 在意义上有些重叠, 但意义不同。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。

文档类	影响命令	cap=true 的默认值	实际定义	cap=false 的默认值	实际定义
article	\section	{}		{}	
book, report	\chapter	\appendixname	附录	\appendixname	Appendix

表 16: appendix/name 选项的默认设置

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。用法与普通章节的 number 选项类似。

该选项也同时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。

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

表 17: appendix/number 选项的默认设置

3.6.5 页面格式设置与汉化

当使用了 CTeX 的文档类或是用 ctex.sty 加载了 heading 选项时,会设置整个文档的页面格式(page style)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,CTeX 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。

CTeX 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。CTeX 宏包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 3.6.4 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 L^AT_EX 页面标记的涵义与使用细节,已经超出了本文档讨论的范围。可以参考 [Knuth(1986c), Chapter 23]、[Mittelbach and Goossens(2004), §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 15

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

而当给 C_TE_X 宏包使用了 fancyhdr 选项时,则会自动调用 fancyhdr 宏包,并设置其 fancy 页面格式使用 \CTEXthechapter 等宏显示中文章节编号。这样,我们可以更为灵活地控制页眉页脚的格式。

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 16

```
\documentclass[fancyhdr]{ctexart}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\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{\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 —

3.7 关于 Lua_AT_EX 下的中文支持方式

在 Lua_AT_EX 下, C_TE_X 宏包依赖 luatexja 宏包来完成中文支持。该宏包是日本 T_EX 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_TE_X 引擎下实现日本 p_TE_X 引擎的(大部分)功能。它为了兼容 p_LA_TE_X 的使用习惯,对 L_AT_EX 2_ε 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,因而 C_TE_X 禁用了它在 L_AT_EX 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 x_eC_JK 宏包大致相同。

3.7.1 Lua[®]TeX 下替代字体的设置

```

AlternateFont \setCJKfamilyfont {<family>}
[
    AlternateFont =
    {
        {<character range1>} [<alternate font features1>] {<alternate font name1>} ||
        {<character range2>} [<alternate font features2>] {<alternate font name2>} ||
        .....
    } ,
    <base font features>
] {<base font name>}

```

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

```

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

```

只设置字体族 *<family>* 在字符范围 *<character range>* 内,对应字形的替代字体。

一个 `\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 17

```

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

```

```

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

```

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

在声明字符范围 *<name>* 的同时,还为 `\setCJKmainfont` 等字体设置命令定义了选项 *<name>*,用于设置对应字符的替代字体:

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

<name> 选项可以与 `AlternateFont` 共同使用,但不能与 `CharRange` 一起使用。如果没有给 *<name>* 设置值,则等价于设置 `CharRange=<name>`,即只设置 *<name>* 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 *<family>* 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。

3.8 杂项

`\CTeX` 用于显示 \CTeX 的标志。

4 对旧版本的兼容性

4.1 \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。

4.2 \CTeX 0.9– \CTeX 1.0d

在 2009 年在 `ctex-kit` 项目成立后, 新增了 \XeTeX 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, 但也部分选项命令功能已失效。

<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	使用 <code>indent</code> 宏包选项会载入 <code>indentfirst</code> 宏包。过时选项。 <code>noindent</code> 什么也不做, 也不会使章节首行停止缩进。需要使用 <code>\ctexset</code> 命令设置章节的 <code>beforekip</code> 选项为正数才能完成原来 <code>noindent</code> 选项的工作。另外, 设置 <code>cap=false</code> 可以保持原有英文文档类的效果, 使章节首行停止缩进。过时选项。
<hr/> zhmap nozhmap <hr/>	<code>zhmap</code> 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 <code>zhmCJK</code> 作为底层中文处理宏包。(3.3.4 节) <code>nozhmap</code> 选项相当于 <code>zhmap=false</code> 。过时选项。
<hr/> winfonts adobefonts nofonts <hr/>	宏包选项 <code>winfonts</code> 相当于 <code>fontset=windows</code> , <code>adobefonts</code> 相当于 <code>fontset=adobe</code> , <code>nofonts</code> 相当于 <code>fontset=none</code> 。这几个选项是过时选项, 对于新文档, 应使用 <code>fontset</code> 选项设置不同字体集。 另外, 第 2 版 \CTeX 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(3.3.4 节)。
<hr/> punct nopunct <hr/>	旧版本中宏包 <code>punct</code> 选项没有参数, 现在可以用参数设定标点风格(3.3.5 节)。原有无参形式的 <code>punct</code> 选项相当于 <code>punct=quanjiao</code> 。 <code>nopunct</code> 选项相当于 <code>punct=plain</code> 。过时选项。
<hr/> cap nocap <hr/>	新版本宏包 <code>cap</code> 选项增加了真假值参数。(3.3.5 节) <code>nocap</code> 选项相当于 <code>cap=false</code> , 成为过时选项。

<u>space</u>	新版本宏包 space 选项增加真假值参数。(3.3.5 节)
<u>nospace</u>	nospace 选项相当于 space=false, 成为过时选项。
<u>fntef</u>	相比旧版本, 新版本宏包选项会禁用 CJKfntef 或 xeCJKfntef 的彩色设置。(3.3.6 节)
<u>\CTEXunderdot</u>	在调用 fntef 宏包选项的同时, 旧版本 CT _E X 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 CT _E X 宏包中已失去意义。 此外, 在 pdfT _E X 引擎下, 用于设置格式的 \CJKunderdotbasesep 等宏也被换名为 \CTEXunderdotbasesep 等宏。 在新版本中, 上述由 fntef 衍生的相关宏都成为过时命令。
<u>\CTEXunderline</u>	
<u>\CTEXunderdblline</u>	
<u>\CTEXunderwave</u>	
<u>\CTEXsout</u>	
<u>\CTEXxout</u>	
<u>\CTEXfilltwosides</u>	
<u>\CTEXindent</u>	更新 \ccwd 宽度后设置 \parindent=2\ccwd。过时命令。
<u>\CTEXnoindent</u>	设置 \parindent=0pt。过时命令。
<u>\CTEXsetup</u>	\CTEXsetup[⟨选项⟩]{⟨标题⟩} 相当于设置了 \ctexset{⟨标题⟩ = {⟨选项⟩}}。过时命令。
<u>\CTEXoptions</u>	\CTEXoptions{⟨选项⟩} 相当于设置了 \ctexset{⟨选项⟩}。过时命令。
<u>captiondelimter</u>	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 CT _E X 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 18

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

4.3 CT_EX 1.02c 以后的 SVN 开发版

CT_EX 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_EX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

5 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)

6 参考文献

- [Knuth(1986c)] DONALD ERVIN KNUTH. *The T_EXbook, Computers & Typesetting*, volume A. Addison-Wesley, 1986c
- [Mittelbach and Goossens(2004)] FRANK MITTELBACH and MICHEL GOOSSENS. *The L^AT_EX Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

7 代码实现

```

1 <@@=ctex>
2 <*ctexcap>
3 \PassOptionsToPackage { heading = true } { ctexcap }
4 \RequirePackageWithOptions { ctex }
5 </ctexcap>
6 <*class|style|ctexsize>
7 \RequirePackage { xparse , l3keys2e }
8 </class|style|ctexsize>
9
10 检查 expl3 和 l3keys2e 的版本。
11 <*class>
12 \msg_new:nnnn { ctex } { l3-too-old }
13 { Support~package~`#1'~too~old. }
14 {
15   Please~update~an~up~to~date~version~of~the~bundles\\
16   `l3kernel'~and~`l3packages'\\
17   using~your~TeX~package~manager~or~from~CTAN.
18 }
19 \@ifpackagelater { expl3 } { 2014/07/20 } { }
20 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
21 \@ifpackagelater { l3keys2e } { 2014/05/05 } { }
22 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
23 </class>
24 <*class|style>
25 \RequirePackage { etoolbox , ifpdf , fix-cm , everyysel }

```

7.1 内部函数与变量

```

\l__ctex_tmp_tl
\g__ctex_tmp_bool 24 \tl_new:N \l__ctex_tmp_tl
\l__ctex_tmp_int 25 \bool_new:N \g__ctex_tmp_bool
\l__ctex_tmp_dim 26 \int_new:N \l__ctex_tmp_int
\l__ctex_tmp_box 27 \dim_new:N \l__ctex_tmp_dim
28 \box_new:N \l__ctex_tmp_box
29 \seq_new:N \l__ctex_tmp_seq

```

(End definition for \l__ctex_tmp_tl and others.)

对旧版本的宏包给出错误信息。

```

30 \msg_new:nnnn { ctex } { package-too-old }
31 { Support~package~`#1'~too~old. }
32 {
33   Please~update~an~up~to~date~version~of~the~package~`#1'\
34   using~your~TeX~package~manager~or~from~CTAN.
35 }

```

\ctex_lua_now_x:n 最新的 expl3 去掉了 l3luatex 模块,因而 \lua_now_x:n 不再有定义。

```

36 \cs_new_eq:NN \ctex_lua_now_x:n \lua_directlua:D

```

(End definition for \ctex_lua_now_x:n.)

\ctex_if_pdfmode:TF \ifpdf 的简单 wrapper。

```

37 \ifpdf
38 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_i:nn
39 \else:
40 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_ii:nn
41 \fi:

```

(End definition for \ctex_if_pdfmode:TF.)

`\ctex_file_input:n` 使用 `\@pushfilename` 和 `\@popfilename` 是为了让文件可以不受当前 L^AT_EX3 语法环境的影响。

```
42 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
43   { \@pushfilename \file_input:n {#1} \@popfilename }
44   \@onlypreamble \ctex_file_input:n
```

(End definition for `\ctex_file_input:n`.)

`\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][]{...}
```

etoolbox 的 `\patchcmd` 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。我们希望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。我们不想依赖 xpatch, 主要是因为它与同作者的 regexprpatch 宏包共用了主要函数的名字, 从而将导致用户不能使用 regexprpatch。

```
45 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
46   { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
47 \group_begin:
48 \char_set_lccode:nn { \< } { \< }
49 \char_set_lccode:nn { \> } { \> }
50 \char_set_lccode:nn { \A } { \t }
51 \tl_map_function:nN { \A \E \S \O \P } \char_set_catcode_other:N
52 \tex_lowercase:D
53   {
54     \group_end:
55     \cs_new_protected:Npn \ctex_parse_name:NNn #1#2#3
56       {
57         \bool_if:nTF { \cs_if_exist_p:c { #3 ~ } || \cs_if_exist_p:c { /#3 } }
58           {
59             \group_begin:
60             \use:x
61             {
62               \__ctex_parse_name:nnNNn { \token_get_replacement_spec:N #2 }
63               \exp_not:N #2 \exp_not:c { #3 ~ } \exp_not:c { /#3 } {#3}
64             } #1
65           }
66         { #1#2 }
67       }
68     \cs_new_protected:Npn \__ctex_parse_name:nnNNn #1#2#3#4#5#6
69       {
70         \group_end:
71         \exp_args:Nc #6
72         {
73           \str_case:nnTF {#1}
74             {
75               { \protect #3 } { }
76               { \x@protect #2 \protect #3 } { }
77             }
78           {
79             \str_if_eq_x:nnTF { \exp_not:n { /@protected@ #3 /#3 } }
80               {
81                 \exp_last_unbraced:Nf \__ctex_parse_name:w
82                 \token_get_replacement_spec:N #3 AESAOPA ~ < \q_stop
83               }
84               { /#5 ~ } { #5 ~ }
```



```

85     }
86     {
87         \str_case:onTF { \_ctex_parse_name:w #1 AESAOPA ~ < \q_stop }
88         {
89             { /@protected@ #2 #4 } { }
90             { /@ #4 } { }
91         }
92         { /#5 } {#5}
93     }
94 }
95 }
96 \cs_new:Npn \_ctex_parse_name:w #1 AESAOPA ~ #2 < #3 \q_stop { #1#2 }
97 }
98 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

(End definition for `\ctex_parse_name:NN`.)

`\ctex_patch_cmd:NnnTF` 在打补丁前先解析实际名字。

```

\ctex_preto_cmd:NnTF 99 \cs_new_protected:Npn \ctex_patch_cmd:NnnTF { \ctex_parse_name:NN \patchcmd }
\ctex_appto_cmd:NnTF 100 \cs_new_protected:Npn \ctex_preto_cmd:NnTF { \ctex_parse_name:NN \pretocmd }
101 \cs_new_protected:Npn \ctex_appto_cmd:NnTF { \ctex_parse_name:NN \apptocmd }

```

(End definition for `\ctex_patch_cmd:NnnTF`, `\ctex_preto_cmd:NnTF`, and `\ctex_appto_cmd:NnTF`.)

`\ctex_patch_cmd:Nnn` 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。并且在补丁的时候关闭 L^AT_EX3 语法。

```

102 \cs_new_protected:Npn \ctex_patch_cmd:Nnn
103 {
104     \group_begin:
105     \char_set_catcode_other:N \#
106     \_ctex_patch_cmd:Nnn
107 }
108 \cs_new_protected:Npn \_ctex_patch_cmd:Nnn #1#2#3
109 {
110     \group_end:
111     \group_begin:
112     \ExplSyntaxOff
113     \ctex_patch_cmd:NnnTF #1 {#2} {#3}
114     {
115         \cs_gset_eq:NN \_ctex_tmp:w #1
116         \group_end:
117         \cs_set_eq:NN #1 \_ctex_tmp:w
118         \cs_undefine:N \_ctex_tmp:w
119     }
120     { \group_end: \ctex_patch_failure:N #1 }
121 }
122 \cs_new_protected:Npn \ctex_patch_failure:N #1
123 { \msg_warning:nxx { ctex } { patch-failure } { \token_to_str:N #1 } }
124 \msg_new:nnn { ctex } { patch-failure }
125 {
126     Patching~command~`#1'~failed.\\
127     ctex~may~not~work~as~expected.
128 }

```

(End definition for `\ctex_patch_cmd:Nnn`.)

`\ctex_default_pt:n` 最新版本的 expl3 已经不允许 `\dim_to_pt:n` 的参数带额外的单位。然而我们需要这个特性实现可展的 `\@defaultunits`。

```

129 \cs_new:Npn \ctex_default_pt:n #1
130 {
131     \exp_after:wN \_ctex_default_pt:w
132     \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
133 }
134 \group_begin:
135 \char_set_catcode_other:N \P
136 \char_set_catcode_other:N \T

```

```

137 \tex_lowercase:D
138 {
139   \group_end:
140   \cs_new:Npn \__ctex_default_pt:w #1 PT #2 \q_stop { #1 PT }
141 }

```

(End definition for \ctex_default_pt:n.)

\l__ctex_encoding_tl (pdf)LaTeX 初始化编码为 GBK, 其它则是 UTF8。

```

142 \tl_new:N \l__ctex_encoding_tl
143 \tl_set:Nx \l__ctex_encoding_tl
144 { \pdfTeX_if_engine:TF { GBK } { UTF8 } }

```

(End definition for \l__ctex_encoding_tl.)

\g__ctex_section_depth_flag 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。

```

145 \cs_new_eq:NN \g__ctex_section_depth_flag \c_two

```

(End definition for \g__ctex_section_depth_flag.)

\g__ctex_zhmCJK_bool 是否使用 zhmCJK 宏包。

```

146 \bool_new:N \g__ctex_zhmCJK_bool

```

(End definition for \g__ctex_zhmCJK_bool.)

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

```

147 \cs_new_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn

```

(End definition for \ctex_zhmap_case:nnn.)

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

```

148 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
149 <style>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfPackage }

```

(End definition for \ctex_at_end:n.)

\g__ctex_std_options_clist 保存传递给标准文档类的选项。

```

150 <*class>
151 \clist_new:N \g__ctex_std_options_clist
152 </class>

```

(End definition for \g__ctex_std_options_clist.)

```

153 </class>style>
154 <*class>style|ctexsize>

```

\g__ctex_font_size_flag 0 表示修改默认字体大小为五号, 1 为小四号, 其它值则不作修改。

```

155 \cs_new_eq:NN \g__ctex_font_size_flag \c_zero

```

(End definition for \g__ctex_font_size_flag.)

对无效选项给出警告。

```

156 \msg_new:nnn { ctex } { invalid-option }
157 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
158 \msg_new:nnn { ctex } { invalid-value }
159 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }

```

对过时选项或命令给出警告。

```

160 \msg_new:nnn { ctex } { deprecated-option }
161 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
162 \msg_new:nnn { ctex } { deprecated-command }
163 { Command~ #1 is~ deprecated.\\ #2 }
164 \msg_new:nnn { ctex } { deprecated-environment }
165 { Environment~`#1'~is~ deprecated.\\ #2 }
166 \cs_generate_variant:Nn \msg_warning:nnnn { nncn }

```

7.2 宏包选项

```
167 \keys_define:nn { ctex / option }
168 {
```

```

c5size
cs4size 169 c5size .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
170 cs4size .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
171 c5size .value_forbidden: ,
172 cs4size .value_forbidden: ,
173 <ctexsize> }
```

(End definition for c5size and cs4size. These functions are documented on page 5.)

```
174 </class|style|ctexsize>
175 <*class|style>
```

linespread 行距初始值为 $1.3 \times 1.2 = 2.56$ 倍字体大小。

```
176 linespread .fp_set:N = \l__ctex_line_spread_fp ,
177 linespread .initial:n = { 1.3 } ,
178 linespread .value_required: ,
```

(End definition for linespread. This function is documented on page 8.)

autoindent 自动调整段落的首行缩进功能。

```
179 autoindent .bool_set:N = \l__ctex_autoindent_bool ,
180 autoindent .initial:n = { true } ,
```

(End definition for autoindent. This function is documented on page 11.)

indent 仅为兼容性保留, 已过时。

```
181 indent .code:n =
182 {
183   \msg_warning:nnn { ctex } { deprecated-option }
184   {
185     The~ indentfirst~ package~ will~ be~ loaded~ but~ the~
186     functionality~ may~ be~ removed~ in~ later~ version.
187     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
188     options.~
189   }
190   \bool_set_true:N \l__ctex_indent_bool
191 } ,
192 indent .value_forbidden: ,
193 noindent .code:n =
194 {
195   \msg_warning:nnn { ctex } { deprecated-option }
196   {
197     The~ functionality~ has~ been~ removed.~
198     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
199     options.~
200   }
201   \bool_set_false:N \l__ctex_indent_bool
202 } ,
203 noindent .value_forbidden: ,
```

(End definition for indent. This function is documented on page 22.)

GBK

```

UTF8 204 GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
205 UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
206 GBK .value_forbidden: ,
207 UTF8 .value_forbidden: ,
```

(End definition for GBK and UTF8. These functions are documented on page 6.)

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、Mac OS X 和其它。

```

208 fontset .tl_gset:N = \g__ctex_fontset_tl ,
209 fontset .value_required: ,
210 nofonts .code:n =
211 {
212   \msg_warning:nnn { ctex } { deprecated-option }
213   {
214     Option~`fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
215     fontset~ option.
216   }
217   \keys_set:nn { ctex / option } { fontset = none }
218 },
219 adobefonts .code:n =
220 {
221   \msg_warning:nnn { ctex } { deprecated-option }
222   {
223     Option~`fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
224     fontset~ option.
225   }
226   \keys_set:nn { ctex / option } { fontset = none }
227 },
228 winfonts .code:n =
229 {
230   \msg_warning:nnn { ctex } { deprecated-option }
231   {
232     Option~`fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
233     fontset~ option.
234   }
235   \keys_set:nn { ctex / option } { fontset = none }
236 },
237 nofonts .value_forbidden: ,
238 winfonts .value_forbidden: ,
239 adobefonts .value_forbidden: ,

```

(End definition for *fontset*. This function is documented on page 7.)

zhmap

```

240 zhmap .choice: ,
241 zhmap .default:n = { true } ,
242 zhmap / zhmcjk .code:n =
243 {
244   \bool_gset_true:N \g__ctex_zhmcjk_bool
245   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
246 },
247 zhmap / true .code:n =
248 {
249   \bool_gset_false:N \g__ctex_zhmcjk_bool
250   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
251 },
252 zhmap / false .code:n =
253 {
254   \bool_gset_false:N \g__ctex_zhmcjk_bool
255   \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
256 },
257 nozhmap .code:n =
258 {
259   \msg_warning:nnn { ctex } { deprecated-option }
260   { Option~`zhmap=false'~ is~ set. }
261   \keys_set:nn { ctex / option } { zhmap = false }
262 },
263 nozhmap .value_forbidden: ,

```

(End definition for *zhmap*. This function is documented on page 22.)

punct 设置标点符号输出格式。

```

264 punct .tl_set:N = \l__ctex_punct_tl ,
265 punct .default:n = { quanjiao } ,
266 punct .initial:n = { quanjiao } ,
267 nopunct .code:n =
268 {
269   \msg_warning:nnn { ctex } { deprecated-option }
270   { Option~`punct=plain'~ is~ set. }
271   \keys_set:nn { ctex / option } { punct = plain }
272 } ,
273 nopunct .value_forbidden: ,

```

(End definition for punct. This function is documented on page 22.)

space

```

274 space .choices:nn =
275 { true , auto , false }
276 {
277   \exp_args:Nx \ctex_at_end:n
278   { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
279 } ,
280 space .default:n = { true } ,
281 nospace .code:n =
282 {
283   \msg_warning:nnn { ctex } { deprecated-option }
284   { Option~`space=false'~ is~ set. }
285   \keys_set:nn { ctex / option } { space = false }
286 } ,
287 nospace .value_forbidden: ,

```

(End definition for space. This function is documented on page 23.)

heading

```

288 heading .bool_set:N = \l__ctex_heading_bool ,

```

(End definition for heading. This function is documented on page 6.)

cap

```

289 cap .bool_set:N = \l__ctex_caption_bool ,
290 cap .initial:n = { true } ,
291 nocap .code:n =
292 {
293   \msg_warning:nnn { ctex } { deprecated-option }
294   { Option~`cap=false'~ is~ set. }
295   \keys_set:nn { ctex / option } { cap = false }
296 } ,
297 nocap .value_forbidden: ,

```

(End definition for cap. This function is documented on page 22.)

sub3section

sub4section

```

298 sub3section .code:n =
299 { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
300 sub4section .code:n =
301 { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
302 sub3section .value_forbidden: ,
303 sub4section .value_forbidden: ,

```

(End definition for sub3section and sub4section. These functions are documented on page 6.)

fntef

fancyhdr

hyperref

```

304 fntef .bool_set:N = \l__ctex_fntef_bool ,
305 fancyhdr .bool_set:N = \l__ctex_fancyhdr_bool ,
306 hyperref .bool_set:N = \l__ctex_hyperref_bool
307 }

```

(End definition for `fntef`, `fancyhdr`, and `hyperref`. These functions are documented on page 23.)

```
308 </class|style>
309 <*class|style|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 310 \clist_map_inline:nn { 10pt , 11pt , 12pt }
311 {
312     \keys_define:nn { ctex / option }
313     {
314         #1 .code:n =
315 <!*class>
316     { \cs_gset_eq:NN \g__ctex_font_size_flag \c_minus_one } ,
317 </!class>
318 <*class>
319     {
320         \cs_gset_eq:NN \g__ctex_font_size_flag \c_minus_one
321         \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
322     } ,
323 </class>
324     #1 .value_forbidden:
325 }
326 }
```

(End definition for 10pt, 11pt, and 12pt. These functions are documented on page 5.)

将未知选项传给标准文档类。

```
327 <*class>
328 \keys_define:nn { ctex / option }
329 {
330     unknown .code:n =
331     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
332 }
333 </class>
334 <|ctexsize>\ctex_file_input:n { ctexopts.cfg }
335 \ProcessKeysOptions { ctex / option }
336 </class|style|ctexsize>
337 <*class|style>
```

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。

```
338 <*class>
339 \if_case:w \g__ctex_font_size_flag
340     \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
341 \or:
342     \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
343 \fi:
```

使用 `\PassOptionsToClass` 是为了预防可能存在的选项冲突。

```
344 <*article>
345 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
346 \LoadClass { article }
347 </article>
348 <*book>
349 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
350 \LoadClass { book }
351 </book>
352 <*report>
353 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
354 \LoadClass { report }
355 </report>
356 </class>
```



```

357 \tl_set_eq:Nc \l__ctex_tmp_tl { ver@ \@currname . \@currentx }
358 <*class>
359 \cs_new_eq:cN { ver@ctex. \@pkgextension } \l__ctex_tmp_tl
360 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \l__ctex_tmp_tl
361 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl
362 </class>
363 <*style>
364 \msg_new:nnnn { ctex } { ctexsize-loaded }
365 { Package~`ctexsize'~can~not~be~loaded~before~`ctex'. }
366 {
367   `ctexsize'~is~actually~a~part~of~`ctex'.\\
368   It~is~not~necessary~to~load~it~separately.
369 }
370 \ifpackageloaded { ctexsize }
371 { \msg_error:nn { ctex } { ctexsize-loaded } }
372 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl }
373 </style>

```

7.3 用户设置接口

\ctexset

```

374 \NewDocumentCommand \ctexset { +m }
375 { \keys_set:nn { ctex } {#1} }

```

(End definition for \ctexset. This function is documented on page 8.)

\CTEXsetup

过时命令。

\CTEXoptions

```

376 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
377 {
378   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
379   { \ctexset~ {ctex/#2}~ {~ #1~ }~ is~ set. }
380   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
381 }
382 \NewDocumentCommand \CTEXoptions { +o }
383 {
384   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
385   { \ctexset~ {ctex}~ {~ #1~ }~ is~ set. }
386   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
387 }

```

(End definition for \CTEXsetup and \CTEXoptions. These functions are documented on page 23.)

7.4 引擎支持

\hypersetup

```

388 \bool_if:NT \l__ctex_hyperref_bool
389 {
390   \cs_if_exist:NF \hypersetup
391   {
392     \cs_new_protected:Npn \hypersetup #1
393     { \PassOptionsToPackage {#1} { hyperref } }
394   }
395   \hypersetup { colorlinks = true }
396   \AtEndPreamble { \RequirePackage { hyperref } }
397 }

```

(End definition for \hypersetup.)

```

398 \pdftex_if_engine:TF
399 {
400   \tl_set:Nx \l__ctex_encoding_tl { \l__ctex_encoding_tl }
401   \ctex_file_input:n { ctex-engine-pdftex.def }
402 }
403 {
404   \tl_set:Nn \l__ctex_encoding_tl { UTF8 }

```

```

405 \xetex_if_engine:TF
406 { \ctex_file_input:n { ctex-engine-xetex.def } }
407 { \ctex_file_input:n { ctex-engine-luatex.def } }
408 }
409 </class|style>
410 <*pdftex>

```

7.4.1 ctex-engine-pdftex.def

首先检查选项,决定是否载入 zhmCJK 宏包。

```

411 \if_bool:N \g__ctex_zhmCJK_bool
412 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
413 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

414 \else:
415 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
416 { \RequirePackage { CJK } }
417 { \RequirePackage { CJKutf8 } }
418 \RequirePackage { CJKpunct , CJKspace }

```

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

```

419 \cs_new_protected_nopar:Npn \ctex_load_zhmap:nnnn #1#2#3#4
420 {
421 \tl_set:Nn \CJKrmdefault {#1}
422 \tl_set:Nn \CJKsfdefault {#2}
423 \tl_set:Nn \CJKttdefault {#3}
424 \AtBeginDvi { \file_input:n {#4} }
425 \AtBeginDocument
426 { \cs_if_exist_use:NT \AtBeginShipoutFirst { { \file_input:n {#4} } } }
427 }
428 \@onlypreamble \ctex_load_zhmap:n

```

(End definition for \ctex_load_zhmap:nnnn.)

```

429 \tl_if_exist:NF \CJKfamilydefault
430 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
431 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
432 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
433 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
434 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
435 { \ctex_patch_failure:N \rmfamily }
436 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
437 { \ctex_patch_failure:N \sffamily }
438 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
439 { \ctex_patch_failure:N \ttfamily }
440 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
441 { \cs_set_eq:NN \reset@font \normalfont }
442 { \ctex_patch_failure:N \normalfont }

```

\CJK@makeActive 应该先于 ctexcap-gbk.cfg 等文件的载入,将汉字的首字节设置为活动字符。使用 zhmCJK 时,此功能已经被启用。

```

443 \CJK@makeActive

```

zhmCJK 判断结束。

```

444 \fi:

```

\ctex_CJK_input:n breqn 包可能会在正文中将 ~ 的 \catcode 改为 12 或 13,这将破坏 CJK 对汉字的首字节的定义(\CJK@loadBinding 和 \CJK@loadEncoding)。因此需要确保载入 .enc 和 .bdg 文件时,~ 的 \catcode 为 7。

```

445 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
446 {
447 \use:x
448 {

```

```

449     \char_set_catcode_other:n      { 60 } % <
450     \char_set_catcode_letter:n     { 64 } % @
451     \char_set_catcode_math_superscript:n { 94 } % ^
452     \int_set_eq:NN \tex_endlinechar:D \c_minus_one
453     \file_input:n {#1}
454     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
455     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
456     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
457     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
458   }
459 }
460 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

(End definition for \ctex_CJK_input:n and \CJK@input.)

\ctex_plane_to_utfxvibe:Nn \CJK@surr 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⁸。我们在这里给出 \CJK@surr 的一个不依赖 \uppercase 的实现。

```

461 \if_cs_exist:N \CJK@surr
462   \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
463   {
464     \tl_set:Nx \l__ctex_tmp_tl {#2}
465     \int_set:Nn \l__ctex_tmp_int
466     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
467     \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
468     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
469     {
470       \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
471       \tl_gset:Nx #1
472       {
473         \int_to_Hex:n
474         { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
475         \int_to_Hex:n
476         { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
477       }
478     }
479   }
480   \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
481 \fi:

```

(End definition for \ctex_plane_to_utfxvibe:Nn and \CJK@surr.)

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

```

482 \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
483 { \AtBeginDocument { \punctstyle { \l__ctex_punct_tl } } }

```

\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}
}

```

⁸<https://github.com/CTeX-org/ctex-kit/issues/146>

```

\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。

```

484 \exp_args:Nx \AtEndPreamble
485 {
486   \exp_not:N \CJK@envStart
487   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
488   \exp_not:N \CJKtilde
489 }

```

\CJK@ignorespaces 默认忽略汉字之间的空格。并关闭名字空间,保存 \CJK@ignorespaces 的定义,方便使用。

```

\ctex_auto_ignorespaces: 490 <@@=>
491 \cs_set_eq:NN \CJK@ignorespaces \CJK@ignorespaces
492 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@ignorespaces

```

恢复名字空间,要把它放在一个 macrocode 环境中,l3doc 才能正确工作。

```

493 <@@=ctex>

```

(End definition for \CJK@ignorespaces and \ctex_auto_ignorespaces:.)

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

```

494 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
495 {
496   \clist_map_inline:Nn \c__ctex_punct_family_clist
497   {
498     \cs_if_free:cF { CJKpunct@ #1 ##1 @spaces }
499     {
500       \cs_set_eq:cc
501       { CJKpunct@ ##1 @spaces }
502       { CJKpunct@ #1 ##1 @spaces }
503     }
504   }
505 }
506 \clist_const:Nn \c__ctex_punct_family_clist
507 {
508   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
509   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
510 }

```

(End definition for \ctex_punct_set:n.)

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

```

511 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
512 {
513   \cs_if_free:cF { CJKpunct@ #2 @spaces }
514   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
515 }

```

(End definition for \ctex_punct_map_family:nn.)

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

```

516 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
517 {
518   \clist_map_inline:nn {#1}
519   {
520     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
521     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
522   }
523 }
524 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
525 {
526   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
527   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}

```

```

528 \CJKpunctmapfamily { C19 } {#1} {#2} { s1 } {#3}
529 \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
530 \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
531 \CJKpunctmapfamily { C70 } {#1} {#2} { s1 } {#3}
532 }

```

(End definition for \ctex_punct_map_bfseries:nn.)

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

```

533 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
534 {
535   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
536   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
537   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
538   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
539   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
540   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
541 }

```

(End definition for \ctex_punct_map_itshape:nn.)

载入边界信息文件。

```

542 \ctex_file_input:n { ctexspa.def }
543 </pdfTeX>
544 <*xetex>

```

7.4.2 ctex-engine-xetex.def

```

545 \RequirePackage { xeCJK }
546 \exp_args:Nx \xeCJKsetup
547 {
548   AutoFakeBold = true ,
549   PunctStyle    = \l__ctex_punct_tl
550 }

```

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。

```

551 \@ifpackagelater { fontspec } { 2014/05/25 } { }
552 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
553 </xetex>
554 <*luatex>

```

7.4.3 ctex-engine-luatex.def

luatexja 为了兼容 p \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。

```

555 \msg_new:nnn { ctex } { luatexja-loaded }
556 {
557   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
558   Loading~file~`#1'~will~abort!
559 }
560 \@ifpackageloaded { luatexja }
561 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
562 { \cs_new_eq:cN { ver@ltj-latex.\@pkgextension } \ExplFileDate }
563 \RequirePackage { luatexja } [ 2013/05/14 ]
564 \RequirePackage { fontspec }
565 \@ifpackagelater { fontspec } { 2014/05/25 } { }
566 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

7.4.3.1 luatexja 的默认设置

567 \ExplSyntaxOff

以下设置抄录自 lltjdefs.sty。

```

568 \ltjdefcharrange{1}{{"80-"36F, "1E00-"1EFF}
569 \ltjdefcharrange{2}{{"370-"4FF, "1F00-"1FFF}
570 \ltjdefcharrange{3}{%
571   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
572 \ltjdefcharrange{4}{%
573   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
574   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
575   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
576 \ltjdefcharrange{5}{{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
577 \ltjdefcharrange{6}{%
578   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
579   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
580 \ltjdefcharrange{7}{
581   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
582   "AC00-"D7FF}
583 \ltjdefcharrange{8}{{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
584 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, +8}}
585 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 ltj-latex.sty。

```

586 \directlua{
587   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
588   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
589 }
590 \ltjsetparameter{kanjiskip=0pt plus 0.4pt minus 0.4pt,
591   xkanjiskip=.25\zw plus 1pt minus 1pt,
592   autospacing, autoxspacing, jacharrange={-1},
593   yalbaselineshift=0pt, yjabaselineshift=0pt,
594   jcharwidowpenalty=500, differentjfm=paverage
595 }
596 \ExplSyntaxOn

```

7.4.3.2 luatexja 的补丁

597 <@@=ctex_ltj>

在 L^AT_EX 下, luatexja 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。

```

598 \RequirePackage { xunicode-addon }
599 \AtBeginUTFCommand
600 {
601   \group_begin:
602   \ctex_lua_now_x:n { tex.globaldefs = 0 }
603   \ltj@allalchar
604 }
605 \AtEndUTFCommand { \group_end: }

```

对 fontspec 沿用 luatexja 的补丁。

606 \RequirePackage { lltjp-fontspec }

lltjp-unicode-math 让数学符号命令成为普通的文字宏。为了避免它被展开, 应该用 \protected 来定义。

```

607 \group_begin:
608 \char_set_catcode_other:n { \c_zero }
609 \cs_new_protected:Npn \__ctex_ltj_um_char:Nw #1 = #2 \q_nil
610 {
611   \group_begin:
612   \char_set_lccode:nn { \c_zero } {#2}
613   \tex_lowercase:D
614   {
615     \group_end:
616     \cs_gset_protected_nopar:Npn #1

```

```

617         {
618             \mode_if_math:TF { ^^@ }
619             { {
620                 \ctex_lua_now_x:n { tex.globaldefs = 0 }
621                 \ltj@allalchar ^^@
622             } }
623         }
624     }
625     \ltjsetmathletter {#2}
626 }
627 \group_end:
628 \AfterPreamble
629 {
630     \cs_if_free:NF \um_cs_set_eq_active_char:Nw
631     { \cs_set_eq:NN \um_cs_set_eq_active_char:Nw \__ctex_ltj_um_char:Nw }
632 }

```

对 listings 的补丁是让代码环境支持 JAchar 类。luatexja 的补丁会将代码目录标题改为日文，我们不需要。

```

633 \AfterPreamble
634 {
635     \@ifpackageloaded { listings }
636     {
637         \use:x
638         {
639             \exp_not:N \RequirePackage { lltp-listings }
640             \tl_set:Nn \exp_not:N \lstlistingname
641             { \exp_not:o { \lstlistingname } }
642             \tl_set:Nn \exp_not:N \lstlistlistingname
643             { \exp_not:o { \lstlistlistingname } }
644         }
645     } { }
646 }

```

7.4.3.3 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 647 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
648 {
649     \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
650     { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
651 }
652 \tl_new:N \CJK@family
653 \tl_new:N \l__ctex_ltj_current_font_tl
654 \tl_set:Nn \l__ctex_ltj_current_font_tl
655 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

(End definition for `\ctex_ltj_select_font:` and `\CJK@family`.)

`__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。

```

656 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
657 {
658     \group_begin:
659     \tl_set_eq:NN \f@encoding \CJK@encoding
660     \tl_set_eq:NN \f@family \CJK@family
661     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
662     \ctex_ltj_pickup_font:
663     \group_end:
664     \font@name
665     \__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。

```

666     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }

```

```

667     { \_ctex_ltj_select_font_aux: }
668   }
669   \cs_new_protected_nopar:Npn \_ctex_ltj_push_fontname:n #1
670   {
671     \cs_gset_eq:NN \_ctex_ltj_save_fontname:w \font@name
672     \cs_gset_nopar:Npx \font@name {#1}
673   }
674   \cs_new_protected_nopar:Npn \_ctex_ltj_pop_fontname:
675   { \cs_gset_eq:NN \font@name \_ctex_ltj_save_fontname:w }

(End definition for \_ctex_ltj_select_font_aux:.)

```

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

```

676   \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
677   {
678     \exp_after:wN \cs_if_exist:NF \font@name
679     {
680       \group_begin:
681         \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
682         \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
683         \define@newfont
684       \group_end:
685     }
686   }
687   \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

(End definition for \ctex_ltj_pickup_font:.)

```

\ctex_ltj_extract_font: luatexja 的 \globaljfont 在 luatexja-core 中定义:

```

%%%%\jfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont{\afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(false)}}
\protected\def\globaljfont{%
\afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(true)}}
\def\ltj@jfont{\directlua{luatexja.jfont.jfontdefY()}}

```

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 '
.. ctemp .. '\endcsname{\ltj@curjfont=' .. fn .. '\relax}')

```

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

```

688   \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
689   {
690     \get@external@font
691     \ctex_ltj_if_alter_shape_exist:nT { \curr@fontshape }
692     {
693       \tl_set:Nx \external@font
694       { \exp_after:wN \_ctex_ltj_patch_external_font:w \external@font }
695     }
696     \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。

```

697     \font@name
698     \ctex_lua_now_x:n { font.current(tex.attribute['ltj@curjfont']) }
699     \use:c { \f@encoding + \f@family }
700     \use:c { \curr@fontshape }
701   }

```

(End definition for \ctex_ltj_extract_font:.)

\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。

```

702 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
703 {
704   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
705   {
706     \group_begin:
707     \tl_set_eq:NN \CJK@family \f@family
708     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
709     {
710       \cs_gset_protected_nopar:Npx \subst@correction
711       {
712         \cs_new_eq:NN
713         \exp_not:c { \l__ctex_ltj_current_font_tl }
714         \font@name
715       }
716       \group_insert_after:N \group_insert_after:N
717       \group_insert_after:N \subst@correction
718     }
719     \group_end:
720   }
721 }

```

(End definition for \ctex_ltj_subst_font:.)

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

```

722 \prg_new_conditional:Npnn \ctex_ltj_if_alternate_shape_exist:n #1 { T , F , TF }
723 {
724   \ctex_lua_now_x:n { luatexja.jfont.does_alt_set (' \luatexluaescapestring {#1}') }
725   \prg_return_true: \else: \prg_return_false: \fi:
726 }

```

(End definition for \ctex_ltj_if_alternate_shape_exist:nTF.)

__ctex_ltj_patch_external_font:w 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 print_aftl_address()
  tex.sprint(cat_lp, ';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。

```

727 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
728 { #1 \ctex_lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for __ctex_ltj_patch_external_font:w.)

\ctex_ltj_select_alternate_font: 在 \selectfont 中更新替代字体。

```

729 \cs_new_protected_nopar:Npn \ctex_ltj_select_alternate_font:
730 {
731   \ctex_ltj_if_alternate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
732   {
733     \ctex_lua_now_x:n
734     {
735       luatexja.jfont.output_alt_font_cmd
736       (' \luatexluaescapestring { \l__ctex_ltj_current_shape_tl }')
737     }

```

```

738         \ctex_lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
739     }
740 }
741 \tl_new:N \l__ctex_ltj_current_shape_tl
742 \tl_set:Nn \l__ctex_ltj_current_shape_tl
743 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for \ctex_ltj_select_alterate_font:.)

\ltj@pickup@altfont@aux 被用在函数 output_alt_font_cmd 中,作用是定义替代字体。

```

744 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@aux #1
745 {
746     \cs_if_exist:cF { #1/\f@size }
747     {
748         \group_begin:
749         \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
750         \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
751         \ctex_ltj_pickup_font:
752         \group_end:
753         \__ctex_ltj_pop_fontname:
754     }
755 }

```

(End definition for \ltj@pickup@altfont@aux.)

756 <@@=>

\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。

```

757 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
758 {
759     \ltj@@getjfontnumber #1
760     \ctex_lua_now_x:n
761     {
762         luatexja.jfont.pickup_alt_font_b
763         ( \the\ltj@tempcntc, '\luatexluaescapestring {#2}' )
764     }
765 }

```

(End definition for \ltj@pickup@altfont@copy.)

766 \ExplSyntaxOff

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

```

767 \def\ltj@@IsFontJapanese#1{%
768     \directlua{luatexja.jfont.is_kenc(string.match(
769         '\luatexluaescapestring{#1}', '[~/+'])}}
770 {\catcode`M=12%
771 \gdef\ltj@@mathJapaneseFonts#1M#2#3\relax{\ltj@@IsFontJapanese{#3}}
772 \let\ltj@@al@getanddefine@fonts=\getanddefine@fonts
773 \def\ltj@@ja@getanddefine@fonts#1#2{%
774     \xdef\font@name{\csname \string#2/\tf@size\endcsname}%
775     \pickup@jfont\let\textfont@name\font@name
776     \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
777     \pickup@jfont\let\scriptfont@name\font@name
778     \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
779     \pickup@jfont
780     \edef\math@fonts{\math@fonts\ltj@setpar@global%
781         \ltj@@set@stackfont#1,\textfont@name:{MJT}}%
782         \ltj@@set@stackfont#1,\scriptfont@name:{MJS}}%
783         \ltj@@set@stackfont#1,\font@name:{MJSS}}%
784     }%
785 }
786 \def\getanddefine@fonts#1#2{%
787     \ltj@tempcnta=#1\ltj@@IsFontJapanese{\string#2}%
788     \ifin\let\ltj@temp=\ltj@@ja@getanddefine@fonts%
789     \else \let\ltj@temp=\ltj@@al@getanddefine@fonts\fi

```

```

790 \ltj@temp{#1}{#2}%
791 }
792 \def\use@mathgroup#1#2{\relax\ifmmode
793 \math@bgroup
794 \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
795 #1\fi\ltj@tempcnta=#2 \expandafter\ltj@@mathJapaneseFonts\string#1\relax%
796 \ifin@jfam#2\relax\else\mathgroup#2\relax\fi
797 \expandafter\math@egroup\fi}%
798 \let\@@italiccorr=\/
799 <@@=ctex_ltj>
800 \ExplSyntaxOn

```

`\g__ctex_ltj_jfm_tl` luatexja 中与标点格式 plain 对应的 JFM 是 mono。

```

801 \tl_new:N \g__ctex_ltj_jfm_tl
802 \tl_gset:Nx \g__ctex_ltj_jfm_tl
803 {
804 \str_if_eq:x:nnTF { \l__ctex_punct_tl } { plain }
805 { mono } { \l__ctex_punct_tl }
806 }

```

(End definition for `\g__ctex_ltj_jfm_tl`.)

`\CJK@encoding` 在 L^AT_EX 下, luatexja 依赖字体编码来实现特殊设置。例如上述的 `\ltj@@IsFontJapanese` 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。

`__ctex_ltj_change_encoding:`

```

807 \tl_const:Nn \CJK@encoding { LTJY3 }
808 \DeclareFontEncoding { \CJK@encoding } { } { }
809 \DeclareFontSubstitution { LTJY3 } { song } { \mddefault } { \updefault }
810 \ctex_lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
811 \cs_new_protected_nopar:Npn \__ctex_ltj_change_encoding:
812 { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
813 \DeclareFontFamily { \CJK@encoding } { song } { }
814 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
815 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\g__ctex_ltj_jfm_tl } { }
816 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
817 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\g__ctex_ltj_jfm_tl } { }
818 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
819 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
820 { \CJK@encoding } { song } { \mddefault } { \updefault }
821 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
822 { \CJK@encoding } { song } { \bfdefault } { \updefault }
823 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
824 \jfam \c__ctex_ltj_math_fam_int

```

(End definition for `\CJK@encoding` and `__ctex_ltj_change_encoding:`.)

7.4.3.4 字体族的定义与使用

这是 luatexja-fontspec 中新增的一些字体选项。

```

825 \newfontfeature { CID } { cid = #1 }
826 \newfontfeature { JFM } { jfm = #1 }
827 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

828 \keys_define:nn { fontspec-preparse-external }
829 {
830 NoEmbed .code:n =
831 { \cs_set_eq:NN \__fontspec_namewrap:n \__ctex_ltj_noembed_wrap:n }
832 }
833 \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。

```

834 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
835 {
836   \group_begin:
837   \clist_clear:N \l__ctex_ltj_char_range_clist
838   \seq_clear:N \l__ctex_ltj_alterate_seq
839   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
840   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
841   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
842   \ctex_ltj_set_alterate_family:nnF {#1} {#3}
843   {
844     \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
845     \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
846       {#1} { \l__ctex_ltj_font_options_clist }
847     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
848     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
849     \__ctex_ltj_gset_family_cs:nn {#1} {#3}
850   }
851   \group_end:
852 }
853 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
854 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
855 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
856 \clist_new:N \l__ctex_ltj_font_options_clist
857 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
858 {
859   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
860   \clist_put_left:Nx #1 { JFM = \g__ctex_ltj_jfm_tl }
861 }

```

(End definition for `\ctex_ltj_set_family:nnn`.)

`\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`

```

862 \prop_new:N \g__ctex_ltj_family_name_prop
863 \prop_new:N \g__ctex_ltj_family_font_name_prop
864 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

(End definition for `\g__ctex_ltj_family_name_prop`, `\g__ctex_ltj_family_font_name_prop`, and `\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。

```

865 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
866 {
867   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
868   {
869     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
870     \cs_undefine:c { \__ctex_ltj_alterate_cs:n {#1} }
871     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
872     {
873       \use:c { \__ctex_ltj_alterate_cs:n { clear / #1 } }
874       \cs_undefine:c { \__ctex_ltj_alterate_cs:n { clear / #1 } }
875       \cs_undefine:c { \__ctex_ltj_alterate_cs:n { reset / #1 } }
876       \prop_gremove:Nn \g__ctex_ltj_reset_alterate_prop {#1}
877     }
878     \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
879   }
880 }
881 \tl_new:N \l__ctex_ltj_tmp_tl
882 \msg_new:nnn { ctex } { redefine-family }
883 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#2). }

```

(End definition for `__ctex_ltj_check_family:n`.)

`__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`。

```

884 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2

```

```

885 {
886   \cs_gset_protected_nopar:cpx { \l__ctex_ltj_family_csname:n {#1} }
887   {
888     \group_begin:
889     \l__ctex_ltj_change_encoding:
890     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
891     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
892     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
893     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
894     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
895     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
896     \exp_not:N \g__ctex_ltj_fontspec_family_tl
897     \l__ctex_ltj_set_alternate_family:n {#1}
898     \group_end:
899   }
900 }
901 \tl_new:N \l__ctex_ltj_base_family_tl
902 \tl_new:N \g__ctex_ltj_fontspec_family_tl
903 \cs_new_nopar:Npn \l__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
904 \cs_new_protected_nopar:Npn \l__ctex_ltj_set_alternate_family:n #1
905 {
906   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
907   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
908   \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n { reset / #1 } }
909   \cs_if_exist_use:c { \l__ctex_ltj_alternate_cs:n {#1} }
910 }
911 \cs_new:Npn \l__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

(End definition for \l__ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

912 \NewDocumentCommand \CJKfamily { m }
913 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
914 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
915 {
916   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
917   {
918     \tl_set:Nn \l__ctex_ltj_family_tl {#1}
919     \selectfont
920   }
921   { \l__ctex_ltj_family_unknown_warning:n {#1} }
922 }
923 \tl_new:N \l__ctex_ltj_family_tl
924 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

(End definition for \CJKfamily.)

\ctex_ltj_family_if_exist:nTF 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。

```

925 \prg_new_protected_conditional:Nppn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
926 {
927   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
928   { \prg_return_true: }
929   {
930     \cs_if_exist_use:cTF { \l__ctex_ltj_family_csname:n {#1} }
931     {
932       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
933       \prg_return_true:
934     }
935     { \prg_return_false: }
936   }
937 }
938 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

_ctex_ltj_family_unknown_warning:n

```

939 \cs_new_protected_nopar:Npn \_ctex_ltj_family_unknown_warning:n #1
940 {
941   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
942   {
943     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
944     {
945       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
946       \msg_warning:nnn { ctex } { family-unknown } {#1}
947     }
948   }
949 }
950 \seq_new:N \g__ctex_ltj_unknown_family_seq
951 \msg_new:nnn { ctex } { family-unknown }
952 {
953   Unknown~CJK~family~\_ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
954   Try~to~use~\_ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
955 }
956 \cs_new_nopar:Npn \_ctex_ltj_msg_def_family_map:n #1
957 {
958   \str_case:x:nnF {#1}
959   {
960     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
961     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
962     \CJKttdefault { \token_to_str:N \setCJKmonofont }
963   }
964   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
965   [...] \{...\}
966 }
967 \cs_new_nopar:Npn \_ctex_ltj_msg_family_map:n #1
968 {
969   \str_case:x:nnF {#1}
970   {
971     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
972     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
973     \CJKttdefault { \token_to_str:N \CJKttdefault }
974   }
975   {#1}
976 }

```

(End definition for _ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

977 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
978 {
979   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
980   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
981   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
982   {
983     \int_gincr:N \g__ctex_ltj_family_int
984     \_ctex_ltj_fontspec:xnn
985     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
986     {#1} {#2}
987   }
988 }
989 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
990 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
991 \cs_new_protected_nopar:Npn \_ctex_ltj_fontspec:nnn #1#2#3
992 {
993   \bool_if:NT \l__ctex_ltj_add_alternate_bool
994   {
995     \cs_if_free:cF
996     { \_ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
997     {
998       \cs_gset_eq:cc
999       { \_ctex_ltj_alternate_cs:n { reset / #1 } }
1000      { \_ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }

```

```

1001         \cs_gset_eq:cc
1002         { \l__ctex_ltj_altername_cs:n { clear / #1 } }
1003         { \l__ctex_ltj_altername_cs:n { clear / \l_ctex_ltj_family_tl } }
1004     }
1005     \bool_set_false:N \l__ctex_ltj_add_altername_bool
1006 }
1007 \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1008 \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1009 \ctex_ltj_switch_family:n {#1}
1010 }
1011 \cs_generate_variant:Nn \l__ctex_ltj_fontspec:nnn { x }
1012 \prop_new:N \g__ctex_ltj_fontspec_prop

```

(End definition for \ctex_ltj_fontspec:nn.)

\ctex_ltj_add_font_features:n

```

\ctex_ltj_add_font_features:nn 1013 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1014 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1015 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1016 {
1017     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1018     {#1} \l__ctex_ltj_tmp_tl
1019     {
1020         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1021         {#1} \l__ctex_ltj_font_options_clist
1022         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1023         \bool_set_true:N \l__ctex_ltj_add_altername_bool
1024         \ctex_ltj_fontspec:xx
1025         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1026         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1027     }
1028     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1029 }
1030 \bool_new:N \l__ctex_ltj_add_altername_bool
1031 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1032 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1033 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1034 {
1035     \token_to_str:N \addCJKfontfeature (s)~ignored.\
1036     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1037 }

```

(End definition for \ctex_ltj_add_font_features:n and \ctex_ltj_add_font_features:nn.)

\setCJKfamilyfont

```

\newCJKfontfamily 1038 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
\CJKfontspec 1039 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1040 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
\addCJKfontfeatures 1041 {
1042     \tl_set:Nx \l__ctex_ltj_tmp_tl
1043     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1044     \cs_new_protected_nopar:Npx #2
1045     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1046     \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1047 }
1048 \NewDocumentCommand \CJKfontspec { O { } m }
1049 {
1050     \ctex_ltj_fontspec:xx {#1} {#2}
1051     \tex_ignorespaces:D
1052 }
1053 \NewDocumentCommand \addCJKfontfeatures { m }
1054 {
1055     \ctex_ltj_add_font_features:x {#1}
1056     \tex_ignorespaces:D
1057 }
1058 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

```

(End definition for \setCJKfamilyfont and others.)

```

\setCJKmainfont
\setCJKsansfont 1059 \NewDocumentCommand \setCJKmainfont { 0 { } m }
1060 {
\setCJKmonofont 1061 \ctex_l tj_set_family:xxx { \CJKrmdefault } {#1} {#2}
\setCJKmathfont 1062 \normalfont
1063 }
\defaultCJKfontfeatures 1064 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1065 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1066 {
1067 \ctex_l tj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1068 \normalfont
1069 }
1070 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1071 {
1072 \ctex_l tj_set_family:xxx { \CJKttdefault } {#1} {#2}
1073 \normalfont
1074 }
1075 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1076 { \ctex_l tj_set_family:xxx { \c__ctex_l tj_math_tl } {#1} {#2} }
1077 \NewDocumentCommand \defaultCJKfontfeatures { m }
1078 { \clist_gset:Nn \g__ctex_l tj_default_features_clist {#1} }
1079 \clist_new:N \g__ctex_l tj_default_features_clist
1080 \@onlypreamble \setCJKmainfont
1081 \@onlypreamble \setCJKsansfont
1082 \@onlypreamble \setCJKmonofont
1083 \@onlypreamble \setCJKmathfont
1084 \@onlypreamble \setCJKromanfont
1085 \@onlypreamble \defaultCJKfontfeatures

(End definition for \setCJKmainfont and others.)

1086 \tl_if_exist:NF \CJKfamilydefault
1087 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1088 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1089 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1090 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1091 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
1092 { \ctex_patch_failure:N \rmfamily }
1093 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
1094 { \ctex_patch_failure:N \sffamily }
1095 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
1096 { \ctex_patch_failure:N \ttfamily }
1097 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
1098 { \cs_set_eq:NN \reset@font \normalfont }
1099 { \ctex_patch_failure:N \normalfont }

```

\ctex_l tj_ensure_default_family: 在导言区结束确认 \CJKfamilydefault 确实存在。

```

1100 \cs_new_protected_nopar:Npn \ctex_l tj_ensure_default_family:
1101 {
1102 \prop_if_empty:NF \g__ctex_l tj_family_font_name_prop
1103 {
1104 \ctex_l tj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_l tj_tmpa_tl
1105 {
1106 \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1107 { \use:n }
1108 {
1109 \ctex_l tj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_l tj_tmpa_tl
1110 { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1111 { \use:n }
1112 }
1113 {
1114 \prop_map_inline:Nn \g__ctex_l tj_family_font_name_prop
1115 {
1116 \prop_map_break:n
1117 { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }

```



```

1118         }
1119     }
1120 }
1121 \normalfont
1122 \ctex_ljt_update_mathfont:
1123 }
1124 }

```

(End definition for \ctex_ljt_ensure_default_family:.)

\ctex_ljt_update_mathfont: 更新数学字体为实际的字体。

```

1125 \cs_new_protected_nopar:Npn \ctex_ljt_update_mathfont:
1126 {
1127     \ctex_ljt_family_if_exist:xNTF { \c__ctex_ljt_math_tl } \l__ctex_ljt_tmp_tl
1128     { \ctex_ljt_update_mathfont:n { \l__ctex_ljt_tmp_tl } }
1129     {
1130         \ctex_ljt_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ljt_tmp_tl
1131         { \ctex_ljt_update_mathfont:n { \l__ctex_ljt_tmp_tl } }
1132     }
1133 }
1134 \cs_new_protected_nopar:Npn \ctex_ljt_update_mathfont:n #1
1135 {
1136     \tl_const:Nx \c__ctex_ljt_math_family_tl {#1}
1137     \DeclareSymbolFont { \c__ctex_ljt_math_tl } { \CJK@encoding }
1138     { \c__ctex_ljt_math_family_tl } { \mddefault } { \updefault }
1139     \cs_if_free:cTF
1140     { \CJK@encoding/\c__ctex_ljt_math_family_tl/\bfdefault/\updefault }
1141     {
1142         \SetSymbolFont { \c__ctex_ljt_math_tl } { bold } { \CJK@encoding }
1143         { \c__ctex_ljt_math_family_tl } { \mddefault } { \updefault }
1144     }
1145     {
1146         \SetSymbolFont { \c__ctex_ljt_math_tl } { bold } { \CJK@encoding }
1147         { \c__ctex_ljt_math_family_tl } { \bfdefault } { \updefault }
1148     }
1149 }

```

(End definition for \ctex_ljt_update_mathfont:.)

7.4.3.5 替代字体的设置

AlternateFont 设置替代字体的选项。

```

CharRange 1150 \keys_define:nn { ctex_ljt / fontspec }
1151 {
1152     AlternateFont .code:n = \ctex_ljt_set_alternate_seq:n {#1} ,
1153     AlternateFont .value_required: ,
1154     CharRange .clist_set:N = \l__ctex_ljt_char_range_clist ,
1155     CharRange .value_required:
1156 }

```

(End definition for AlternateFont and CharRange. These functions are documented on page 21.)

\ctex_ljt_set_alternate_seq:n 我们使用 || 作为替代字体序列的分隔标志。它可能被设置为活动字符,为此需要先“消毒”,同时过滤掉空元素。

```

1157 \group_begin:
1158 \char_set_catcode_active:N \/
1159 \char_set_lccode:nn { `\/ } { `| }
1160 \tex_lowercase:D
1161 {
1162     \group_end:
1163     \cs_new_protected:Npn \ctex_ljt_set_alternate_seq:n #1
1164     {
1165         \clist_if_empty:NT \l__ctex_ljt_char_range_clist
1166         {
1167             \tl_set:Nn \l__ctex_ljt_tmp_tl {#1}

```

```

1168         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { // } { || }
1169         \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1170         \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1171         { ! \tl_if_blank:n { ##1 } }
1172         \seq_concat:Nnn \l__ctex_ltj_alternate_seq
1173         \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1174     }
1175 }
1176 }
1177 \seq_new:N \l__ctex_ltj_tmp_seq
1178 \seq_new:N \l__ctex_ltj_alternate_seq

(End definition for \ctex_ltj_set_alternate_seq:n)

```

`\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。

```

1179 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1180 {
1181     \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1182     {
1183         \__ctex_ltj_check_family:n {#1}
1184         \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1185         { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1186         #3
1187     }
1188     { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1189 }

(End definition for \ctex_ltj_set_alternate_family:nnF)

```

`\ctex_ltj_save_alternate_seq:Nn` 保存由 AlternateFont 设置的替代字体序列。

```

\ctex_ltj_save_alternate_seq:Nnnwn 1190 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1191 {
1192     \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1193     { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1194 }
1195 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1196 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1197 { m m m +O{ } m u{ \q_stop } }
1198 {
1199     \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1200     \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1201     \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1202     \tl_if_blank:nTF {#5}
1203     { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1204     {
1205         \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1206         \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1207     }
1208     \use:x
1209     {
1210         \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1211         { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1212         { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1213         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1214     }
1215 }
1216 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

(End definition for \ctex_ltj_save_alternate_seq:Nn and \ctex_ltj_save_alternate_seq:Nnnwn.)

`\ctex_ltj_set_alternate_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。

```

1217 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nn #1#2
1218 {
1219     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1220     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist

```

```

1221 \ctex_ltj_set_alternate_family:coonn
1222 { \__ctex_ltj_alternate_cs:n {#1} }
1223 { \l__ctex_ltj_char_range_clist }
1224 { \l__ctex_ltj_font_options_clist } {#2} {#1}
1225 }
1226 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1227 {
1228   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1229   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1230   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1231 }
1232 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

(End definition for \ctex_ltj_set_alternate_family:nn)

```

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

```

1233 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1234 {
1235   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1236   \cs_gset_protected_nopar:Npx #1
1237   { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1238 }

(End definition for \ctex_ltj_save_alternate_family:Nnnn)

```

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

```

1239 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1240 {
1241   \group_begin:
1242   \__ctex_ltj_change_encoding:
1243   \cs_set_eq:NN \CJKfamily \use_none:n
1244   \ctex_ltj_swap_cs:NN
1245   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1246   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1247   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1248   \group_end:
1249 }
1250 \tl_new:N \l__ctex_ltj_alternate_family_tl

(End definition for \ctex_ltj_set_alternate_family:nnn)

```

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1251 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1252 {
1253   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1254   \cs_set_eq:NN #1 #2
1255   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1256   \cs_undefine:N \__ctex_ltj_tmp:w
1257 }

(End definition for \ctex_ltj_swap_cs:NN)

```

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。

`__ctex_ltj_update_family_uid:N`

```

1258 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1259 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1260 {
1261   \int_gincr:N \g__ctex_ltj_family_int
1262   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1263 }
1264 \int_new:N \g__ctex_ltj_family_int

(End definition for LTJFONTUID and \__ctex_ltj_update_family_uid:N)

```

`\ctex_ltj_declare_altername_shape:nnnnnn` 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 $\langle series \rangle$ 和第四个参数 $\langle shape \rangle$ 。

```

1265 \cs_new_protected:Npn \ctex_ltj_declare_altername_shape:nnnnnn #1#2#3#4#5#6
1266 {
1267   \ctex_ltj_declare_altername_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1268   \ctex_ltj_set_altername_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1269   { \l__ctex_ltj_base_family_tl } {#3} {#4}
1270   { \l_fontspec_family_tl } {#3} {#4}
1271 }

```

(End definition for `\ctex_ltj_declare_altername_shape:nnnnnn`.)

`\ctex_ltj_set_altername_shape:Nnnnnnn` 与 `luatexja` 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。

```

1272 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:Nnnnnnn #1#2#3#4#5#6#7
1273 {
1274   \clist_map_inline:Nn #1
1275   {
1276     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1277     {
1278       \ctex_ltj_set_altername_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1279       \l__ctex_ltj_char_range_tl
1280     }
1281     { \ctex_ltj_set_altername_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1282   }
1283   \__ctex_ltj_save_altername_shape:cnn
1284   { \__ctex_ltj_altername_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1285   { luatexja.jfont.clear_alt_font_latex }
1286   { '\luatexluaescapestring { \CJK@encoding/#2/#3/#4 }' }
1287 }

```

(End definition for `\ctex_ltj_set_altername_shape:Nnnnnnn`.)

`\ctex_ltj_set_altername_shape:nnn` 我们使用 `->` 而不是像 `luatexja` 一样使用 `-` 作为区间的分隔符。`luatexja` 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。

```

1288 \NewDocumentCommand \ctex_ltj_set_altername_shape:nnn
1289 { m m > { \SplitArgument { \c_one } { -> } } m }
1290 { \ctex_ltj_set_altername_shape:nnnn {#1} {#2} #3 }
1291 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:nnnn #1#2#3#4
1292 {
1293   \ctex_ltj_set_altername_shape:n
1294   {
1295     \IfNoValueTF {#4}
1296     { \int_eval:n {#3} , \int_eval:n {#3} , }
1297     {
1298       \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,
1299       \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1300     }
1301     '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1302     '\luatexluaescapestring { \CJK@encoding/#1 }'
1303   }
1304 }
1305 \cs_new_protected_nopar:Npn \ctex_ltj_set_altername_shape:n #1
1306 {
1307   \ctex_lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1308   \__ctex_ltj_save_altername_shape:cnn
1309   { \__ctex_ltj_altername_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1310   { luatexja.jfont.set_alt_font_latex } {#1}
1311 }

```

(End definition for `\ctex_ltj_set_altername_shape:nnn`.)

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

```

1312 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1313 {
1314   \tl_map_inline:Nn #3
1315   {
1316     \ctex_ltj_set_alternate_shape:n
1317     {
1318       ##1 ,
1319       '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1320       '\luatexluaescapestring { \CJK@encoding/#1 }'
1321     }
1322   }
1323 }

```

(End definition for `\ctex_ltj_set_alternate_shape:nnN`)

`_ctex_ltj_save_alternate_shape:Nnn` 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。

```

1324 \cs_new_protected_nopar:Npn \_ctex_ltj_save_alternate_shape:Nnn #1#2#3
1325 {
1326   \group_begin:
1327   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1328   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1329   \cs_set_eq:NN \luatexluaescapestring \scan_stop:
1330   \cs_gset_protected_nopar:Npx #1
1331   { \exp_not:o {#1} \exp_not:N \ctex_lua_now_x:n { #2 ( #3 ) } }
1332   \group_end:
1333 }
1334 \cs_generate_variant:Nn \_ctex_ltj_save_alternate_shape:Nnn { c }

```

(End definition for `_ctex_ltj_save_alternate_shape:Nnn`)

`clearalternatefont` 清除和重置操作总是全局的。

```

1335 \keys_define:nn { ctex }
1336 {
1337   clearalternatefont .code:n =
1338   { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1339   resetalternatefont .code:n =
1340   { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1341   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1342   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1343 }
1344 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1345 {
1346   \group_begin:
1347   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1348   {
1349     \cs_if_exist_use:cT { \_ctex_ltj_alternate_cs:n { clear / #1 } }
1350     {
1351       \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1352       {#1} { \l__ctex_ltj_base_family_tl }
1353       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1354       \selectfont
1355     }
1356   }
1357   { \_ctex_ltj_family_unknown_warning:n {#1} }
1358   \group_end:
1359 }
1360 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alternate_font:n #1
1361 {
1362   \group_begin:
1363   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1364   {
1365     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1366     \use:c { \_ctex_ltj_alternate_cs:n { reset / #1 } }

```

```

1367         \selectfont
1368     }
1369     \group_end:
1370 }
1371 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1372 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalternatefont` and `resetalternatefont`. These functions are documented on page 21.)

declarecharrange 预先声明字符范围。

```

1373 \keys_define:nn { ctex }
1374 {
1375     declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1376     declarecharrange .value_required:
1377 }
1378 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1379 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1380 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1381 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1382 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

(End definition for `declarecharrange`. This function is documented on page 21.)

```

\ctex_ltj_declare_char_range:nn #1 是名字,#2 是范围。
\g__ctex_ltj_char_range_prop 1383 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1384 {
1385     \tl_clear:N \l__ctex_ltj_char_range_tl
1386     \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1387     \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1388     \ctex_ltj_def_char_range_key:n {#1}
1389     \tl_clear:N \l__ctex_ltj_char_range_tl
1390 }
1391 \tl_new:N \l__ctex_ltj_char_range_tl
1392 \prop_new:N \g__ctex_ltj_char_range_prop

```

(End definition for `\ctex_ltj_declare_char_range:nn` and `\g__ctex_ltj_char_range_prop`.)

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

```

1393 \NewDocumentCommand \ctex_ltj_save_char_range:n
1394 { > { \SplitArgument { \c_one } { -> } } m }
1395 { \ctex_ltj_save_char_range:nn #1 }
1396 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1397 {
1398     \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1399     { {
1400         \IfNoValueTF {#2}
1401         { \int_eval:n {#1} , \int_eval:n {#1} }
1402         {
1403             \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1404             \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1405         }
1406     } }
1407 }

```

(End definition for `\ctex_ltj_save_char_range:n`.)

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

```

1408 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1409 {
1410     \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1411     {
1412         \keys_define:nn { ctex_ltj / fontspec }
1413         { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1414     }
1415 }

```

(End definition for \ctex_ltj_def_char_range_key:n.)

\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。

```

1416 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1417 {
1418   \tl_if_blank:nTF {#2}
1419   { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1420   {
1421     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1422     {
1423       \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1424       \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1425     }
1426   }
1427 }

```

(End definition for \ctex_ltj_char_range_key:nn.)

__ctex_ltj_char_range_parse_feature:w 可以使用加方括号的方式,通过文件名来调用字体。这容易与字体选项混淆。例如,需要将 [simsum.ttc] 设置为 range 的主字体,就需要使用 range={{[simsum.ttc]}} 或者 []{[simsum.ttc]}。下面的目的是,支持直接使用 [simsum.ttc] 和 [...] [simsum.ttc]。

```

1428 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1429 { +o o u { \q_stop } }
1430 {
1431   \exp_args:Nnf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1432   {
1433     \IfNoValueTF {#1} { {#3} }
1434     {
1435       \IfNoValueTF {#2}
1436       { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ { #1 } ] { #3 } } }
1437       { [ { #1 } ] { { #2 } } }
1438     }
1439   }
1440   \seq_put_right:No \l__ctex_ltj_alterate_seq { \l__ctex_ltj_tmp_tl }
1441 }

```

(End definition for __ctex_ltj_char_range_parse_feature:w.)

7.4.3.6 其它设置

\ctex_ltj_update_xkanjiskip: \ltjsetparameter 对 xkanjiskip 是即时赋值。 \zw 与字体相关,因此需要每次 \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,最后的设置会影响全段。

```

1442 \cs_new_protected_nopar:Npn \ctex_ltj_update_xkanjiskip:
1443 {
1444   \skip_if_eq:nnT
1445   { \ltjgetparameter { xkanjiskip } } { \l__ctex_ltj_xkanjiskip_skip }
1446   {
1447     \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1448     \ltjsetparameter { xkanjiskip = \l__ctex_ltj_xkanjiskip_skip }
1449   }
1450 }

```

注意,此时还没有实际设置字体,所以 \zw 是 0pt。

```

1451 \skip_new:N \l__ctex_ltj_xkanjiskip_skip
1452 \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }

```

(End definition for \ctex_ltj_update_xkanjiskip: and \l__ctex_ltj_xkanjiskip_skip.)

在抄录环境中禁用 autospacing 和 autoxspacing。然而,luatexja 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
}

1453 \AtBeginDocument
1454 {
1455   \ctex_appto_cmd:NnTF \verbatim@font { \CTEX@verbatim@font@hook }
1456   { } { \ctex_patch_failure:N \verbatim@font }
1457 }
1458 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1459 { \ltjsetparameter { autospacing = false , autoxspacing = false } }
1460 <@@=ctex>
1461 </luatex>
1462 <*pdfTeX|xetex|luatex>

```

7.4.4 调整 \CJKfamilydefault

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。

```

1463 <*pdfTeX|luatex>
1464 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1465 {
1466   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1467   {
1468     \group_begin:
1469     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1470     \tl_gset:Nx \CJKfamilydefault
1471     {
1472       \str_case:onF { \familydefault }
1473       {
1474         { \rmdefault } { \exp_not:N \CJKrmdefault }
1475         { \sfdefault } { \exp_not:N \CJKsfdefault }
1476         { \ttdefault } { \exp_not:N \CJKttdefault }
1477       }
1478       { \CJKfamilydefault }
1479     }
1480     \group_end:
1481   }

```

使用 Lua[®]TeX 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 宏包时, C19rm 等总是有定义的, 不需要确认。

```

1482 <*luatex>
1483   \ctex_ljtj_ensure_default_family:
1484 </luatex>
1485 }
1486 \AtEndPreamble { \ctex_update_default_family: }

```

(End definition for \ctex_update_default_family:.)

\l__ctex_family_default_init_tl 往 \CJKfamilydefault 中加入标志, 用于判断它是否被更改。

```

1487 \tl_new:N \l__ctex_family_default_init_tl
1488 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1489 \tl_set:Nx \l__ctex_family_default_init_tl
1490 {
1491   \exp_not:N \__ctex_family_default_wrap:n
1492   { \exp_not:o { \CJKfamilydefault } }
1493 }

```



```

1494 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1495 </pdf|luatex>

(End definition for \l__ctex_family_default_init_tl.)

```

7.4.5 操作系统的判断

`\ctex_detected_platform:` 在 LuaTeX 下直接用调用 `os.name` 来判断。

```

1496 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1497 <*luatex>
1498 {
1499   \tl_gset:Nx \g__ctex_fontset_tl
1500   {
1501     \ctex_lua_now_x:n
1502     {
1503       if ~ os.name == 'windows' then ~
1504         tex.sprint ( 'windows' )
1505       elseif ~ os.name == 'macosx' then ~
1506         tex.sprint ( 'mac' )
1507       else ~
1508         tex.sprint ( 'fandol' )
1509       end
1510     }
1511   }
1512 }
1513 </luatex>

```

pdfTeX 和 XeTeX 下则依据 `/dev/null` 和 `nul:` 的存在性以及文件系统的大小写敏感性来判断。Mac OS X 的大小写敏感性在安装时是可选的。为了保险起见,这里的判断很繁琐,最多要进行 4 次文件操作!

```

1514 <*xetex|pdf|luatex>
1515 {
1516   \file_if_exist:nTF { /dev/null }
1517   {
1518     \file_if_exist:nTF { nul: }
1519     {
1520       \file_if_exist:nTF { \c__ctex_engine_file_tl }
1521       { \ctex_if_macosx:TF { mac } { windows } }
1522       { \ctex_if_macosx:TF { mac } { fandol } }
1523     }
1524     { \ctex_if_macosx:TF { mac } { fandol } }
1525   }
1526   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1527 }
1528 \tex_uppercase:D \exp_after:wN
1529 {
1530   \exp_after:wN \tl_const:Nn \exp_after:wN \c__ctex_engine_file_tl
1531   \exp_after:wN { \g_file_current_name_tl }
1532 }

```

(End definition for `\ctex_detected_platform:`.)

`\ctex_if_macosx:TF` 以 `/mach_kernel` 为特征文件判断 Mac OS X。

```

1533 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1534 {
1535   \file_if_exist:nTF { \c__ctex_macosx_file_tl }
1536   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1537   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1538 }
1539 \tl_const:Nn \c__ctex_macosx_file_tl { /mach_kernel }
1540 </xetex|pdf|luatex>

```

(End definition for `\ctex_if_macosx:TF`.)

7.4.6 hyperref 的支持选项

处理各个引擎下的 PDF 中文书签问题。在 pdfTeX 下使用 GBK 编码, DVIPDFMx 驱动可以直接用它的 \special 命令, 其它模式用 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。

```

1541 <*pdfTeX>
1542 \cs_if_free:NF \hypersetup
1543 {
1544   \hypersetup { driverfallback = dvipdfmx }
1545   \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1546   {
1547     \hypersetup { CJKbookmarks = true }
1548     \ctex_if_pdfmode:TF
1549     { \RequirePackage { xCJK2uni } }
1550     {
1551       \AtEndPreamble
1552       {
1553         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1554         {
1555           \AtBeginShipoutFirst
1556           { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1557         }
1558         { \RequirePackage { xCJK2uni } }
1559       }
1560     }
1561   }
1562   { \hypersetup { pdfencoding = unicode } }
1563 }
1564 </pdfTeX>

```

在 XeTeX 下, hyperref 在处理带有非 ASCII 字符和 \% 的书签时有问题⁹。事实上, hyperref 在驱动文件 hxeTeX.def 中设置了 \Hy@unicodetrue, 从而书签总是会被 \HyPsd@ConvertToUnicode 转化成 UTF-16BE 编码的形式(抄录自 \pdfstringdef 的定义):

```

\ifHy@unicode
\HyPsd@ConvertToUnicode#1%
\ifx\HyPsd@pdfencoding\HyPsd@pdfencoding@auto
\ltx@ifundefined{StringEncodingConvertTest}{%
}{%
\edef\unescapestring\HyPsd@temp#1%
\ifxetex
\let\HyPsd@UnescapedString\HyPsd@temp
\StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
{utf16be}{ascii-print}{%
\edef\escapestring\HyPsd@temp\HyPsd@temp
\global\let#1\HyPsd@temp
\HyPsd@EscapeTeX#1%
\Hy@unicodetrue
}{%
\HyPsd@ToBigChars#1%
}%

```

通过宏包选项 pdfencoding=unicode 设置 \HyPsd@pdfencoding 为 unicode, 可以避免随后再将书签从 UTF-16BE 字节流转化回正常字符(其中使用的 \HyPsd@ToBigChars 没有考虑书签中含有 \% 的情况)。Heiko Oberdiek 在 README 中说明了将书签转化回正常字符的意图: 避免 XDVI PDFMx 的警告¹⁰:

```
** WARNING ** Failed to convert input string to UTF16...
```

XeTeX 的维护者 Khaled Hosny 已经注意到了这个问题¹¹。需要注意的是, hxeTeX.def 重载了宏包选项 unicode, 目的是不能设置它为 false, 但也导致它不会改变 \HyPsd@pdfencoding。

⁹<https://github.com/CTeX-org/ctex-kit/issues/39>

¹⁰<http://project.ktug.org/dvipdfmx/mailman/dvipdfmx/2009-December/000153.html>

¹¹<http://tug.org/pipermail/tex-live/2013-December/034613.html>

如果 `hyperref` 先于 `CTEX` 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 `XgTeX` 下更保险的 `pdfencoding` 选项来设置。为了与 `XgTeX` 下的行为一致 (使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 `LuaTeX` 下也启用这个选项。

```

1565 <*xetex|luatex>
1566 \cs_if_exist_use:NT \hypersetup { { pdfencoding = unicode } }
1567 </xetex|luatex>

```

7.4.7 fntef 选项

载入 `CJKfntef` 或 `xeCJKfntef` 并做适当格式设置。有关 `\CTEX` 开头的宏定义是过时命令, 仅做兼容性保留。

```

1568 <*luatex>
1569 \msg_new:nnn { ctex } { fntef-not-available }
1570 { Option~`fntef'~is~not~available~in~LuaLaTeX. }
1571 </luatex>
1572 \if_bool:N \l__ctex_fntef_bool
1573 <pdfTeX> \RequirePackage { CJKfntef } \normaleM
1574 <xetex> \RequirePackage { xeCJKfntef }
1575 <luatex> \msg_warning:nn { ctex } { fntef-not-available }
1576 \clist_map_inline:nn
1577 { underdot , underline , underdblline , underwave , sout , xout }
1578 <*pdfTeX|xetex>
1579 {
1580 <pdfTeX> \cs_set_eq:cN { CJK#1color } \c_empty_tl
1581 <xetex> \keys_set:nn { xeCJK / options } { #1 / format = { } }
1582 \cs_new_protected_nopar:cpx { CTEX#1 }
1583 {
1584 \exp_not:n
1585 {
1586 \msg_warning:nnnn { ctex } { deprecated-command } { CTEX#1 }
1587 { You~ can~ use~ the~ command~ with~ prefix~ \CJK~ instead. }
1588 }
1589 \exp_not:c { CJK#1 }
1590 }
1591 }
1592 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1593 {
1594 \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1595 { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1596 \CJKfilltwosides
1597 }
1598 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1599 </pdfTeX|xetex>
1600 <*luatex>
1601 { \cs_new_eq:cN { CTEX#1 } \use:n }
1602 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1603 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1604 </luatex>
1605 <*pdfTeX>
1606 \clist_map_inline:nn
1607 {
1608 underdotbasesep , underdotsep , underlinebasesep ,
1609 underlinesep , underdbllinesep , underdbllinebasesep ,
1610 underwavebasesep , underwavesep , southeight ,
1611 underdotcolor , underwavecolor , underlinecolor ,
1612 underdbllinecolor , soutcolor , xoutcolor
1613 }
1614 {
1615 \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1616 \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1617 }
1618 </pdfTeX>
1619 \fi:

```

7.4.8 \ccwd 的更新

`\ctex_update_ccwd:`

```

\ccwd 1620 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1621 {*pdfTeX|xetex}
1622 {
1623   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1624   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1625 }
1626 {/pdfTeX|xetex}
1627 {*luatex}
1628 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1629 {/luatex}
1630 \dim_new:N \ccwd

(End definition for \ctex_update_ccwd: and \ccwd.)

```

`\ctex_update_ccglue:` 更新字间距。

```

1631 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1632 {*pdfTeX}
1633 {
1634   \cs_set_protected_nopar:Npn \CJKglue
1635   { \skip_horizontal:N \l__ctex_ccglue_skip }
1636 }
1637 {/pdfTeX}
1638 {*xetex}
1639 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1640 {/xetex}
1641 {*luatex}
1642 { \ltjsetparameter { kanjiskip = \l__ctex_ccglue_skip } }
1643 {/luatex}
1644 \skip_new:N \l__ctex_ccglue_skip

(End definition for \ctex_update_ccglue:.)

```

`\ctex_if_ccglue_touched_p:` 检查用户是否修改过汉字间距。

```

\ctex_if_ccglue_touched:TF 1645 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1646 {
1647   {*pdfTeX|xetex}
1648   \if_meaning:w \CJKglue \__ctex_ccglue:
1649   \prg_return_false: \else: \prg_return_true: \fi:
1650 {/pdfTeX|xetex}
1651 {*luatex}
1652   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1653   { \prg_return_false: } { \prg_return_true: }
1654 {/luatex}
1655 }

```

注意下面的标记不能用 `%<pdfTeX|xetex>`，它会导致旧版本的 `l3docstrip` 不能替换 `@@`。

```

1656 {*pdfTeX|xetex}
1657 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }
1658 {/pdfTeX|xetex}

(End definition for \ctex_if_ccglue_touched:TF.)

```

`\ctex_update_em_unit:` 将当前汉字的宽度保存到 `\ccwd` 中备用。不采用 1 em，因为这时的 1 em 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，`luatexja` 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。

```

1659 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1660 {pdfTeX|xetex} { \dim_set:Nn \ccwd { \f@size \p@ } }
1661 {luatex} { \dim_set_eq:NN \ccwd \zw }

```

(End definition for \ctex_update_em_unit:.)

7.4.9 其它

\ctex_add_to_selectfont:n \EverySelectfont 直到文档开始时才有效。为了 \ccwd 和 luatexja 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。everyisel 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。

```

1662 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1663 {
1664   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1665   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1666 }
1667 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1668 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1669 \use:n
1670 {
1671   \ExplSyntaxOff
1672   \ctex_preto_cmd:NnTF \selectfont { \CTEX@selectfont@hook }
1673   {
1674     \tl_put_left:Nn \@EverySelectfont@Init
1675     { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1676   }
1677   { \ctex_patch_failure:N \selectfont }
1678 }
1679 \ExplSyntaxOn

```

(End definition for \ctex_add_to_selectfont:n and \CTEX@selectfont@hook.)

\CJK@plane 有定义, 说明处于 CJK 宏包的 \CJKsymbol 之内, 不必使用钩子。

```

1680 <*pdfTeX>
1681 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1682 </pdfTeX>
1683 <*xetex|luatex>
1684 \EverySelectfont { \CTEX@selectfont@hook }
1685 </xetex|luatex>

```

Attribute 寄存器 \ltj@curjfnt 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。

```

1686 <*luatex>
1687 \ctex_add_to_selectfont:n
1688 {
1689   \ctex_ltj_select_font:
1690   \ctex_ltj_select_alternate_font:
1691   \ctex_ltj_update_xkanjiskip:
1692 }
1693 \tl_set:Nn \CJK@family { song } \selectfont
1694 \tl_clear:N \CJK@family
1695 </luatex>

```

space 在正文中设置忽略空格方式。

```

1696 \keys_define:nn { ctex }
1697 {
1698   <*pdfTeX|xetex>
1699     space .choice: ,
1700     space .default:n = { true } ,
1701     space / true .code:n =
1702     <pdfTeX> { \cs_set_eq:NN \CJK@ignorespaces \prg_do_nothing: } ,
1703     <xetex> { \xeCJKsetup { CJKspace = true } } ,
1704     space / auto .code:n =
1705     <pdfTeX> { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: } ,
1706     <xetex> { \xeCJKsetup { CJKspace = false } } ,
1707     space / false .code:n =
1708     <pdfTeX> { \cs_set_eq:NN \CJK@ignorespaces \tex_ignorespaces:D }
1709     <xetex> { \xeCJKsetup { CJKspace = false } }

```

```

1710 </pdf|tex|xetex>
1711 <(*luatex)
1712     space .code:n =
1713     { \msg_warning:nn { ctex } { invalid-option } }
1714 </luatex>
1715 }

```

(End definition for space. This function is documented on page 23.)

```

1716 </pdf|tex|xetex|luatex>
1717 <(*class|style|ctexsize>

```

7.5 中文字号

\zihao

```

1718 \NewDocumentCommand \zihao { m }
1719 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

(End definition for \zihao. This function is documented on page 9.)

\ctex_zihao:n

```

1720 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
1721 {
1722     \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
1723     { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
1724     { \msg_error:nnn { ctex } { fontsize } {#1} }
1725 }
1726 \msg_new:nnnn { ctex } { fontsize }
1727 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
1728 {
1729     The~old~font~size~is~used~if~you~continue.\\
1730     The~available~font~sizes~are~listed~as~follow.\\
1731     \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
1732 }

```

(End definition for \ctex_zihao:n.)

7.5.1 定义中文字号

\c__ctex_font_size_prop 基础行距是字号的 1.2 倍,采用 ϵ -TeX 的 scaling 运算得到的结果要比简单的 $1.2\dimexpr$ 精确¹²。

__ctex_save_font_size:nn

```

1733 \prop_new:N \c__ctex_font_size_prop
1734 \seq_new:N \c__ctex_font_size_seq
1735 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
1736 {
1737     \use:x
1738     {
1739         \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
1740         {
1741             { \dim_to_decimal:n {#2} }
1742             { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
1743         }
1744     }
1745     \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
1746 }
1747 \clist_map_inline:nn
1748 {
1749     { 8 } { 5 bp } ,
1750     { 7 } { 5.5 bp } ,
1751     { -6 } { 6.5 bp } ,
1752     { 6 } { 7.5 bp } ,
1753     { -5 } { 9 bp } ,
1754     { 5 } { 10.5 bp } ,

```

¹²<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

1755 { -4 } { 12 bp } ,
1756 { 4 } { 14 bp } ,
1757 { -3 } { 15 bp } ,
1758 { 3 } { 16 bp } ,
1759 { -2 } { 18 bp } ,
1760 { 2 } { 22 bp } ,
1761 { -1 } { 24 bp } ,
1762 { 1 } { 26 bp } ,
1763 { -0 } { 36 bp } ,
1764 { 0 } { 42 bp }
1765 }
1766 { \__ctex_save_font_size:nn #1 }

```

(End definition for \c__ctex_font_size_prop.)

7.5.2 修改默认字号大小

\ctex_set_font_size:Nnn

```

1767 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
1768 {
1769   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
1770   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
1771   { \msg_error:nnn { ctex } { fontsize } {#2} }
1772 }
1773 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
1774 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

(End definition for \ctex_set_font_size:Nnn.)

```

1775 \if_case:w \g__ctex_font_size_flag
1776 \ctex_set_font_size:Nnn \normalsize { 5 }
1777 {
1778   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
1779   \abovedisplayshortskip \z@ \@plus3\p@
1780   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
1781   \belowdisplayskip \abovedisplayskip
1782   \let\@listi\@listI
1783 }
1784 \ctex_set_font_size:Nnn \small { -5 }
1785 {
1786   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
1787   \abovedisplayshortskip \z@ \@plus2\p@
1788   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
1789   \def\@listi{\leftmargin\leftmarginI
1790     \topsep 4\p@ \@plus2\p@ \@minus2\p@
1791     \parsep 2\p@ \@plus\p@ \@minus\p@
1792     \itemsep \parsep}
1793   \belowdisplayskip \abovedisplayskip
1794 }
1795 \ctex_set_font_size:Nnn \footnotesize { 6 }
1796 {
1797   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
1798   \abovedisplayshortskip \z@ \@plus\p@
1799   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
1800   \def\@listi{\leftmargin\leftmarginI
1801     \topsep 3\p@ \@plus\p@ \@minus\p@
1802     \parsep 2\p@ \@plus\p@ \@minus\p@
1803     \itemsep \parsep}
1804   \belowdisplayskip \abovedisplayskip
1805 }
1806 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
1807 \ctex_set_font_size:Nnn \tiny { 7 } { }
1808 \ctex_set_font_size:Nnn \large { -4 } { }
1809 \ctex_set_font_size:Nnn \Large { -3 } { }
1810 \ctex_set_font_size:Nnn \LARGE { -2 } { }
1811 \ctex_set_font_size:Nnn \huge { 2 } { }
1812 \ctex_set_font_size:Nnn \Huge { 1 } { }

```

```

1813 \or:
1814 \ctex_set_font_size:Nnn \normalsize { -4 }
1815 {
1816   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
1817   \abovedisplayshortskip \z@ \@plus3\p@
1818   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
1819   \belowdisplayskip \abovedisplayskip
1820   \let\@listi\@listI
1821 }
1822 \ctex_set_font_size:Nnn \small { 5 }
1823 {
1824   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
1825   \abovedisplayshortskip \z@ \@plus3\p@
1826   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
1827   \def\@listi{\leftmargin\leftmargini
1828             \topsep 9\p@ \@plus3\p@ \@minus5\p@
1829             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
1830             \itemsep \parsep}
1831   \belowdisplayskip \abovedisplayskip
1832 }
1833 \ctex_set_font_size:Nnn \footnotesize { -5 }
1834 {
1835   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
1836   \abovedisplayshortskip \z@ \@plus3\p@
1837   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
1838   \def\@listi{\leftmargin\leftmargini
1839             \topsep 6\p@ \@plus2\p@ \@minus2\p@
1840             \parsep 3\p@ \@plus2\p@ \@minus\p@
1841             \itemsep \parsep}
1842   \belowdisplayskip \abovedisplayskip
1843 }
1844 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
1845 \ctex_set_font_size:Nnn \tiny { -6 } { }
1846 \ctex_set_font_size:Nnn \large { -3 } { }
1847 \ctex_set_font_size:Nnn \Large { -2 } { }
1848 \ctex_set_font_size:Nnn \LARGE { 2 } { }
1849 \ctex_set_font_size:Nnn \huge { -1 } { }
1850 \ctex_set_font_size:Nnn \Huge { 1 } { }
1851 \fi:

```

\ctex_declare_math_sizes:nnnn

```

1852 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
1853 {
1854   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
1855   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
1856 }

```

(End definition for \ctex_declare_math_sizes:nnnn.)

__ctex_get_font_sizes:Nn

```

1857 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
1858 {
1859   \tl_clear:N #1
1860   \tl_map_inline:nn {#2}
1861   {
1862     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
1863     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
1864     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
1865   }
1866 }

```

(End definition for __ctex_get_font_sizes:Nn.)

```

1867 \clist_map_inline:nn
1868 {
1869   { 8 } { 8 } { 5pt } { 5pt } ,
1870   { 7 } { 7 } { 5pt } { 5pt } ,

```



```

1871 { -6 }{ -6 }{ 5pt }{ 5pt } ,
1872 { 6 }{ 6 }{ 5pt }{ 5pt } ,
1873 { -5 }{ -5 }{ 6pt }{ 5pt } ,
1874 { 5 }{ 5 }{ 7pt }{ 5pt } ,
1875 { -4 }{ -4 }{ 8pt }{ 6pt } ,
1876 { 4 }{ 4 }{ 5 }{ 6 } ,
1877 { -3 }{ -3 }{ -4 }{ -5 } ,
1878 { 3 }{ 3 }{ 4 }{ 5 } ,
1879 { -2 }{ -2 }{ -3 }{ -4 } ,
1880 { 2 }{ 2 }{ 3 }{ 4 } ,
1881 { -1 }{ -1 }{ -2 }{ -3 } ,
1882 { 1 }{ 1 }{ 2 }{ 3 } ,
1883 { -0 }{ -0 }{ -1 }{ -2 } ,
1884 { 0 }{ 0 }{ 1 }{ 2 }
1885 }
1886 { \ctex_declare_math_sizes:nnnn #1 }
1887 <ctexsize>\normalsize
1888 </class|style|ctexsize>
1889 <*class|style>

```

7.5.3 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。

```

1890 \keys_define:nn { ctex }
1891 {
1892   autoindent .choice: ,
1893   autoindent .default:n = { true } ,
1894   autoindent / true .code:n =
1895   {
1896     \bool_set_true:N \l__ctex_autoindent_bool
1897     \ctex_update_parindent:
1898   } ,
1899   autoindent / false .code:n =
1900   { \bool_set_false:N \l__ctex_autoindent_bool }
1901 }

```

(End definition for autoindent. This function is documented on page 11.)

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

1902 \NewDocumentCommand \CTEXsetfont { }
1903 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

(End definition for \CTEXsetfont. This function is documented on page 9.)

\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。

```

1904 \cs_new_protected_nopar:Npn \ctex_update_size:
1905 {
1906   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
1907   {
1908     \ctex_update_stretch:
1909     \ctex_update_parindent:
1910   }
1911   { \ctex_update_ziju: }
1912 }
1913 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
1914 \tl_new:N \l__ctex_ziju_tl
1915 \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。

```

1916 \ctex_add_to_selectfont:n
1917 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

(End definition for \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`。

```

1918 \keys_define:nn { ctex }
1919 {
1920   linestretch .code:n =
1921   {
1922     \dim_compare:nNnTF
1923     { \ctex_default_pt:n {#1} } = { \ctex_default_pt:n { #1 in } }
1924     { \tl_set:Nn \l__ctex_line_stretch_tl {#1} }
1925     { \tl_set:Nn \l__ctex_line_stretch_tl { #1 \ccwd } }
1926     \CTEXsetfont
1927   } ,
1928   linestretch .value_required:
1929 }
1930 \tl_new:N \l__ctex_line_stretch_tl
1931 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

(End definition for `linestretch`. This function is documented on page 11.)

\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。 ε -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。

```

1932 \cs_new_protected_nopar:Npn \ctex_update_stretch:
1933 {
1934   \ctex_update_em_unit:
1935   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
1936   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
1937   {
1938     \ctex_if_ccglue_touched:TF
1939     { \ctex_update_ccwd: }
1940     {
1941       \dim_set:Nn \l__ctex_tmp_dim
1942       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
1943       \skip_set:Nn \l__ctex_ccglue_skip
1944       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
1945       \ctex_update_ccglue:
1946     }
1947   }
1948   {
1949     \int_set:Nn \l__ctex_tmp_int
1950     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
1951     \tex_divide:D \l__ctex_tmp_int \ccwd
1952     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
1953     {
1954       \skip_set:Nn \l__ctex_ccglue_skip
1955       {
1956         \c_zero_dim plus \dim_eval:n
1957         {
1958           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
1959           \l__ctex_tmp_int
1960         }
1961       }
1962     }
1963     { \skip_zero:N \l__ctex_ccglue_skip }
1964     \ctex_update_ccglue:
1965   }
1966 }

```

(End definition for `\ctex_update_stretch:`.)

`\ctex_update_parindent:` 更新段落首行缩进。此函数在字号变化时调用。

```

1967 \cs_new_protected_nopar:Npn \ctex_update_parindent:
1968 {
1969     \bool_if:NT \l__ctex_autoindent_bool
1970     {
1971         \dim_compare:nNf \parindent = \c_zero_dim
1972         { \dim_set:Nn \parindent { 2 \ccwd } }
1973     }
1974 }

```

(End definition for `\ctex_update_parindent:`.)

`\ziju` 若参数为 0, 则恢复正常间距。

```

1975 \NewDocumentCommand \ziju { m }
1976 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
1977 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
1978 {
1979     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
1980     \CTEXsetfont
1981 }

```

(End definition for `\ziju`. This function is documented on page 9.)

`\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。

```

1982 \cs_new_protected_nopar:Npn \ctex_update_ziju:
1983 {
1984     \ctex_update_em_unit:
1985     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
1986     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
1987     \dim_compare:nNfTF \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 决定伸展还是收缩。

```

1988     {
1989         \dim_set:Nn \l__ctex_tmp_dim
1990         { \linewidth - \ccwd + \l__ctex_ziju_dim }
1991         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
1992         \tex_divide:D \l__ctex_tmp_int \ccwd
1993         \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。

```

1994         \dim_compare:nNf \parindent = \c_zero_dim
1995         {
1996             \int_compare:nNf \l__ctex_tmp_int < \c_three
1997             { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
1998         }
1999         \skip_set:Nn \l__ctex_ccglue_skip
2000         {
2001             \l__ctex_ziju_dim
2002             plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2003             minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2004             { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
2005         }
2006     }
2007     { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2008     \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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $2\backslash\mathrm{ccwd}$ 。

```

2009     \ctex_update_parindent:
2010 }
2011 \dim_new:N \l__ctex_ziju_dim

```

(End definition for \ctex_update_ziju:.)

更新行距。

```
2012 \linespread { \fp_use:N \l__ctex_line_spread_fp }
```

激活默认字体大小,更新 \parindent 和 \CJKglue。

```
2013 \normalsize
```

\footnotesep 由于我们加宽了行距,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。

```
2014 \group_begin: \footnotesize \exp_args:NNNo \group_end:
```

```
2015 \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
```

(End definition for \footnotesep.)

\CTEXindent 过时命令。

```
\CTEXnoindent 2016 \NewDocumentCommand \CTEXindent { }
2017 {
2018   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
2019   { \parindent is~ set~ to~ 2\ccwd. }
2020   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
2021 }
2022 \NewDocumentCommand \CTEXnoindent { }
2023 {
2024   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2025   { \parindent is~ set~ to~ 0pt. }
2026   \dim_zero:N \parindent
2027 }
```

(End definition for \CTEXindent and \CTEXnoindent. These functions are documented on page 23.)

```
2028 \bool_if:NT \l__ctex_indent_bool
```

```
2029 { \RequirePackage { indentfirst } }
```

7.6 中文数字与日期

```
2030 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
```

```
2031 \RequirePackage { zhnumber }
```

\chinese

```
2032 \cs_new_eq:NN \chinese \zhnum
2033 \cs_new_eq:NN \Chinese \chinese
2034 \cs_new_eq:NN \CTEXcounter \use_none:n
```

(End definition for \chinese. This function is documented on page 11.)

\CTEXnumber

```
\CTEXdigits 2035 \NewDocumentCommand \CTEXnumber { m m }
2036 { \protected@edef #1 { \zhnumber {#2} } }
2037 \NewDocumentCommand \CTEXdigits { m m }
2038 { \protected@edef #1 { \zhdigits {#2} } }
```

(End definition for \CTEXnumber and \CTEXdigits. These functions are documented on page 11.)

today

```
2039 \cs_set_eq:NN \CTEX@todayold \today
2040 \keys_define:nn { ctex }
2041 {
2042   today .choice: ,
2043   today / old .code:n =
2044     { \cs_set_eq:NN \today \CTEX@todayold } ,
2045   today / small .code:n =
2046     {
2047       \cs_set_eq:NN \today \zhtoday
```

```

2048     \zhnumsetup { time = Arabic }
2049   } ,
2050   today / big .code:n =
2051   {
2052     \cs_set_eq:NN \today \zhtoday
2053     \zhnumsetup { time = Chinese }
2054   } ,
2055   today / unknown .code:n =
2056   { \msg_error:nxx { ctex } { today-undef } {#1} }
2057 }
2058 \msg_new:nnnn { ctex } { today-undef }
2059 { Today~format~`#1'~is~undefined. }
2060 { Available~today~formats~are~`old',~`small',~and~`big'. }
2061 \bool_if:NT \l__ctex_caption_bool
2062 { \keys_set:nn { ctex } { today = small } }

```

(End definition for today. This function is documented on page 12.)

7.7 其它中文标题定义

```

2063 \keys_define:nn { ctex }
2064 {
2065   contentsname .tl_set:N = \contentsname ,
2066   listfigurename .tl_set:N = \listfigurename ,
2067   listtablename .tl_set:N = \listtablename ,
2068   figurename .tl_set:N = \figurename ,
2069   tablename .tl_set:N = \tablename ,
2070   abstractname .tl_set:N = \abstractname ,
2071   indexname .tl_set:N = \indexname ,
2072   appendixname .tl_set:N = \appendixname ,
2073   <article> bibname .tl_set:N = \refname
2074   <book|report> bibname .tl_set:N = \bibname
2075 }
2076 <*style>
2077 \msg_new:nnn { ctex } { ctexbibname }
2078 {
2079   Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.\\
2080   The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2081 }
2082 \tl_if_exist:NTF \bibname
2083 { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2084 {
2085   \tl_if_exist:NTF \refname
2086   { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2087   {
2088     \msg_warning:nn { ctex } { ctexbibname }
2089     \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2090   }
2091 }
2092 </style>
2093 </class|style>
2094 <*class|heading>

```

7.8 中文化的标题结构

7.8.1 定义标题格式选项

\c__ctex_headings_seq

```

2095 \seq_new:N \c__ctex_headings_seq
2096 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2097 {
2098   <article> part , section , subsection , subsubsection ,
2099   <book|report> part , chapter , section , subsection , subsubsection ,
2100   paragraph , subparagraph
2101 }

```

(End definition for \c__ctex_headings_seq.)

__ctex_initial_heading:n

```

2102 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2103 {
2104   \tl_new:c { CTEX@pre#1 }
2105   \tl_new:c { CTEX@post#1 }
2106   \tl_const:cx { CTEX@the#1 }
2107   {
2108     \exp_not:c { CTEX@pre#1 }
2109     \exp_not:c { CTEX@the#1 }
2110     \exp_not:c { CTEX@post#1 }
2111   }
2112   \tl_const:cx { CTEX@#1name }
2113   {
2114     \exp_not:c { CTEX@#1nameformat }
2115     \exp_not:c { CTEX@pre#1 }
2116     \exp_not:N \tl_if_empty:NTF \exp_not:c { CTEX@#1numberformat }
2117     { \exp_not:c { CTEX@the#1 } }
2118     {
2119       \group_begin:
2120         \exp_not:c { CTEX@#1numberformat }
2121         \exp_not:c { CTEX@the#1 }
2122       \group_end:
2123     }
2124     \exp_not:c { CTEX@post#1 }
2125     \exp_not:c { CTEX@#1aftername }
2126   }
2127 }

```

(End definition for __ctex_initial_heading:n.)

__ctex_def_heading_keys:n

```

2128 \cs_new_protected_nopar:Npn \__ctex_def_heading_keys:n #1
2129 {
2130   \tl_put_right:Nx \l__ctex_tmp_tl
2131   {
2132     #1 .meta:nn = { ctex / #1 } { ##### } ,
2133     #1 / name .code:n =
2134       { \ctex_assign_heading_name:nn {#1} { ##### } } ,
2135     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2136     #1 / format .tl_set:N = \exp_not:c { CTEX@#1format } ,
2137     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1nameformat } ,
2138     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1numberformat } ,
2139     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1aftername } ,
2140     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1titleformat } ,
2141     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1beforeskip } ,
2142     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1afterskip } ,
2143     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1indent } ,
2144     #1 / format+ .code:n =
2145       { \tl_put_right:Nn \exp_not:c { CTEX@#1format } { ##### } } ,
2146     #1 / nameformat+ .code:n =
2147       { \tl_put_right:Nn \exp_not:c { CTEX@#1nameformat } { ##### } } ,
2148     #1 / numberformat+ .code:n =
2149       { \tl_put_right:Nn \exp_not:c { CTEX@#1numberformat } { ##### } } ,
2150     #1 / aftername+ .code:n =
2151       { \tl_put_right:Nn \exp_not:c { CTEX@#1aftername } { ##### } } ,
2152     #1 / titleformat+ .code:n =
2153       { \tl_put_right:Nn \exp_not:c { CTEX@#1titleformat } { ##### } } ,
2154     #1 / beforeskip .initial:n = \c_zero_skip ,
2155     #1 / afterskip .initial:n = \c_zero_skip ,
2156     #1 / indent .initial:n = \c_zero_dim ,
2157     #1 / beforeskip .value_required: ,
2158     #1 / afterskip .value_required: ,
2159     #1 / indent .value_required: ,
2160   }
2161 }

```

(End definition for _ctex_def_heading_keys:n.)

`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^AT_EX3 的 `clist` 总是会自动忽略空元素，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。

```

2162 \NewDocumentCommand \ctex_assign_heading_name:nn
2163 { m > { \SplitArgument { \c_one } { , } } +m }
2164 { \_ctex_assign_heading_name:nnn {#1} #2 }
2165 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2166 {
2167   \tl_set:cn { CTEX@pre#1 } {#2}
2168   \IfNoValueTF {#3}
2169     { \tl_clear:c { CTEX@post#1 } }
2170     { \tl_set:cn { CTEX@post#1 } {#3} }
2171 }

(End definition for \ctex_assign_heading_name:nn and \_ctex_assign_heading_name:nnn.)

2172 \tl_clear:N \l__ctex_tmp_tl
2173 \seq_map_inline:Nn \c__ctex_headings_seq
2174 {
2175   \_ctex_initial_heading:n {#1}
2176   \_ctex_def_heading_keys:n {#1}
2177 }
2178 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }
2179 \tl_clear:N \l__ctex_tmp_tl

```

`pagestyle` 只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2180 <*book|report>
2181 \keys_define:nn { ctex }
2182 {
2183   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2184   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle
2185 }
2186 </book|report>

```

(End definition for `pagestyle`. This function is documented on page ??.)

7.8.2 标准标题命令的修改

7.8.2.1 `part` 的标题

2187 <@@=>

`\part`

```

2188 <*article>
2189 \renewcommand\part{%
2190   \if@noskipsec \leavevmode \fi
2191   \par
2192   % \addvspace{4ex}%
2193   \@tempskipa \CTEX@part@beforeskip \relax
2194   \ifdim \@tempskipa <\z@
2195     \@tempskipa -\@tempskipa \@afterindentfalse
2196   \else
2197     \@afterindenttrue
2198   \fi
2199   \addvspace{\@tempskipa}%
2200   \secdef\@part\@spart}
2201 </article>

```

(End definition for `\part`.)

\@part

```

2202 (*article)
2203 \def\@part[#1]#2{%
2204   \ifnum \c@secnumdepth >\m@ne
2205     \refstepcounter{part}%
2206     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2207     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2208   \else
2209     \addcontentsline{toc}{part}{#1}%
2210   \fi
2211   {\interlinepenalty \@M
2212   % \normalfont \parindent \z@ \raggedright
2213   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2214   \ifnum \c@secnumdepth >\m@ne
2215     % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2216     \CTEX@partname
2217   \fi
2218   % \huge\bfseries #2%
2219   \CTEX@part@titleformat{#2}%
2220   \markboth{}{}\par}%
2221   \nobreak
2222   % \vskip 3ex
2223   \vskip \CTEX@part@afterskip
2224   \@afterheading}
2225 \end{article}

```

标准文档类是在 \@part 和 \@chapter 定义的最开始设置 \thispagestyle, 我们这里的修改出现在它之后, 可以覆盖之前的设置。

```

2226 (*book|report)
2227 \def\@part[#1]#2{%
2228   \thispagestyle{\CTEX@part@pagestyle}%
2229   \ifnum \c@secnumdepth >-2\relax
2230     \refstepcounter{part}%
2231     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2232     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2233   \else
2234     \addcontentsline{toc}{part}{#1}%
2235   \fi
2236   \markboth{}{}\par}
2237   {\interlinepenalty \@M
2238   % \normalfont \centering
2239   \normalfont \CTEX@part@format
2240   \ifnum \c@secnumdepth >-2\relax
2241     % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2242     \CTEX@partname
2243   \fi
2244   % \Huge\bfseries #2\par}%
2245   \CTEX@part@titleformat{#2}\par}%
2246   \@endpart}
2247 \end{book|report}

```

(End definition for \@part.)

\@spart

```

2248 (*article)
2249 \def\@spart#1{%
2250   {\interlinepenalty \@M
2251   % \normalfont \parindent \z@ \raggedright
2252   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2253   % \huge \bfseries #1\par}%
2254   \CTEX@part@titleformat{#1}\par}%
2255   \nobreak
2256   % \vskip 3ex
2257   \vskip \CTEX@part@afterskip
2258   \@afterheading}
2259 \end{article}

```



```

2260 <*book|report>
2261 \def\@spart#1{%
2262     {\interlinepenalty \@M
2263      % \normalfont \centering
2264       \normalfont \CTEX@part@format
2265      % \Huge \bfseries #1\par}%
2266      \CTEX@part@titleformat{#1}\par}%
2267     \@endpart}
2268 </book|report>

```

(End definition for \@spart.)

7.8.2.2 chapter 的标题

```

2269 <*book|report>

\@chapter
2270 \def\@chapter[#1]#2{%
2271     \ifnum \c@secnumdepth >\m@ne
2272     <book>     \if@mainmatter
2273                 \refstepcounter{chapter}%
2274                 % \typeout{\@chapapp\space\thechapter.}%
2275                 \typeout{\CTEXthechapter}%
2276                 \addcontentsline{toc}{chapter}
2277                 % {\protect\numberline{\thechapter}#1}%
2278                 {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2279     <book>     \else
2280     <book>     \addcontentsline{toc}{chapter}{#1}%
2281     <book>     \fi
2282     \else
2283         \addcontentsline{toc}{chapter}{#1}%
2284         \fi
2285         \chaptermark{#1}%
2286         \addtocontents{lof}{\protect\addvspace{10\p@}}%
2287         \addtocontents{lot}{\protect\addvspace{10\p@}}%
2288         \if@twocolumn
2289             \topnewpage[\@makechapterhead{#2}]%
2290         \else
2291             \@makechapterhead{#2}%
2292         \@afterheading
2293         \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2294 \def\@makechapterhead#1{%
2295     \thispagestyle{\CTEX@chapter@pagestyle}%
2296     % \vspace*{50\p@}%
2297     \@tempskipa \CTEX@chapter@beforeskip \relax
2298     \ifdim \@tempskipa <\z@
2299         \@tempskipa -\@tempskipa \@afterindentfalse
2300     \else
2301         \@afterindenttrue
2302     \fi
2303     \vspace*{\@tempskipa}%
2304     % {\normalfont \parindent \z@ \raggedright
2305      % {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2306       \ifnum \c@secnumdepth >\m@ne
2307       <book>     \if@mainmatter
2308           % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2309           \CTEX@chaptername
2310       <book>     \fi
2311       \fi
2312       \interlinepenalty\@M
2313       % \Huge \bfseries #1\par\nobreak
2314       \CTEX@chapter@titleformat{#1}\par\nobreak
2315       % \vskip 40\p@

```

```

2316 \vskip \CTEX@chapter@afterskip
2317 }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2318 \def\@makeschapterhead#1{%
2319 \thispagestyle{\CTEX@chapter@pagestyle}%
2320 % \vspace*{50\p@}%
2321 \@tempskipa \CTEX@chapter@beforeskip \relax
2322 \ifdim \@tempskipa <\z@
2323 \@tempskipa -\@tempskipa \@afterindentfalse
2324 \else
2325 \@afterindenttrue
2326 \fi
2327 \vspace*{\@tempskipa}%
2328 % {\normalfont \parindent \z@ \raggedright
2329 {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2330 \interlinepenalty\@M
2331 % \Huge \bfseries #1\par\nobreak
2332 \CTEX@chapter@titleformat{#1}\par\nobreak
2333 % \vskip 40\p@
2334 \vskip \CTEX@chapter@afterskip
2335 }}

```

(End definition for \@makeschapterhead.)

```

2336 \</book|report>

```

7.8.2.3 section 类的标题

\@secntformat

```

2337 \def\@secntformat#1{%
2338 \ifundefined{CTEX@#1name}%
2339 {\csname the#1\endcsname\quad}%
2340 {\csname CTEX@#1name\endcsname}}

```

(End definition for \@secntformat.)

\@sect

```

2341 \def\@sect#1#2#3#4#5#6[#7]#8{%
2342 \ifnum #2>\c@secnumdepth
2343 \let\@svsec\@empty
2344 \else
2345 \refstepcounter{#1}%
2346 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2347 \fi
2348 \@tempskipa #5\relax
2349 \ifdim \@tempskipa>\z@
2350 \begingroup
2351 #6{%
2352 \@hangfrom{\hskip #3\relax\@svsec}%
2353 % \interlinepenalty \@M #8\@@par}%
2354 \interlinepenalty \@M
2355 \csname CTEX@#1@titleformat\endcsname{#8}\@@par}%
2356 \endgroup
2357 \csname #1mark\endcsname{#7}%
2358 \addcontentsline{toc}{#1}{%
2359 \ifnum #2>\c@secnumdepth \else
2360 % \protect\numberline{\csname the#1\endcsname}%
2361 \protect\numberline{\ifundefined{CTEXthe#1}%
2362 {\csname the#1\endcsname}%
2363 {\csname CTEXthe#1\endcsname}}}%
2364 \fi
2365 #7}%
2366 \else

```

```

2367 \def\@svsechd{%
2368 #6{\hskip #3\relax
2369 % \@svsec #8}%
2370 \@svsec \csname CTEX@#1@titleformat\endcsname{#8}}%
2371 \csname #1mark\endcsname{#7}%
2372 \addcontentsline{toc}{#1}{%
2373 \ifnum #2>\c@secnumdepth \else
2374 % \protect\numberline{\csname the#1\endcsname}%
2375 \protect\numberline{\@ifundefined{CTEXthe#1}%
2376 {\csname the#1\endcsname}%
2377 {\csname CTEXthe#1\endcsname}}}%
2378 \fi
2379 #7}}%
2380 \fi
2381 \@xsect{#5}}

```

(End definition for \@sect.)

\@ssect 并没有参数给出当前标题的名字, 扩展它的参数会与 hyperref 冲突。它的第二个参数是 BEFORESKIP(\@startsection 的第四个参数), 在定义中并没有被用到, 应该可以用它来传递名字。我们这里通过函数 \CTEX@titleformat@n 来传递, 它将在 #4 中被重定义为相应的 titleformat。

```

2382 \def\@ssect#1#2#3#4#5{%
2383 \@tempskipa #3\relax
2384 \ifdim \@tempskipa>\z@
2385 \begingroup
2386 #4{%
2387 \@hangfrom{\hskip #1}%
2388 % \interlinepenalty \@M #5\@par}%
2389 \interlinepenalty \@M
2390 \CTEX@titleformat@n{#5}\@par}%
2391 \endgroup
2392 \else
2393 % \def\@svsechd{#4{\hskip #1\relax #5}}%
2394 \def\@svsechd{#4{\hskip #1\relax \CTEX@titleformat@n{#5}}}%
2395 \fi
2396 \@xsect{#3}}

```

(End definition for \@ssect.)

```
2397 <@@=ctex>
```

\CTEX@set@titleformat@n 在 \@startsection 中设置 \CTEX@titleformat@n 为相应函数。

```

\CTEX@titleformat@n 2398 \cs_new_protected_nopar:Npn \CTEX@set@titleformat@n #1
2399 { \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat } }
2400 \cs_new_eq:NN \CTEX@titleformat@n \use:n

```

(End definition for \CTEX@set@titleformat@n and \CTEX@titleformat@n.)

```

2401 \int_zero:N \l__ctex_tmp_int
2402 \clist_map_inline:nn
2403 { section , subsection , subsubsection , paragraph , subparagraph }
2404 {
2405 \int_incr:N \l__ctex_tmp_int
2406 \cs_gset_protected_nopar:cpx {#1}
2407 {
2408 \exp_not:N \@startsection {#1}
2409 { \int_use:N \l__ctex_tmp_int }
2410 { \exp_not:c { CTEX@#1@indent } }
2411 { \exp_not:c { CTEX@#1@before skip } }
2412 { \exp_not:c { CTEX@#1@after skip } }
2413 {
2414 \CTEX@set@titleformat@n {#1}
2415 \exp_not:N \normalfont \exp_not:c { CTEX@#1@format }
2416 }
2417 }
2418 }

```

7.8.2.4 附录标题

```

2419 \tl_new:N \CTEX@preappendix
2420 \tl_new:N \CTEX@postappendix
2421 \keys_define:nn { ctex }
2422 { appendix .meta:nn = { ctex / appendix } {#1} }
2423 \keys_define:nn { ctex / appendix }
2424 {
2425     name .code:n = { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2426     name .initial:n = { \appendixname \space } ,
2427     number .tl_set:N = \CTEX@appendixnumber ,
2428     <article> number .initial:n = { \@Alph \c@section }
2429     <book|report> number .initial:n = { \@Alph \c@chapter }
2430 }

```

\appendix

```

2431 \cs_new_eq:NN \CTEX@save@appendix \appendix
2432 \cs_gset_protected_nopar:Npn \appendix
2433 {
2434     \CTEX@save@appendix
2435     <*article>
2436     \gdef \CTEX@presection { \CTEX@preappendix }
2437     \gdef \CTEX@thesection { \CTEX@appendixnumber }
2438     \gdef \CTEX@postsection { \CTEX@postappendix }
2439 </article>
2440 <*book|report>
2441     \gdef \CTEX@prechapter { \CTEX@preappendix }
2442     \gdef \CTEX@thechapter { \CTEX@appendixnumber }
2443     \gdef \CTEX@postchapter { \CTEX@postappendix }
2444 </book|report>
2445 }

```

(End definition for \appendix.)

7.8.3 目录标签的宽度

\numberline

```

2446 \cs_new_protected:Npn \CTEX@toc@width@n #1
2447 {
2448     \hbox_set:Nn \l__ctex_tmp_box {#1}
2449     \dim_set:Nn \@tempdima
2450     {
2451         \dim_max:nn { \@tempdima }
2452         { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2453     }
2454 }
2455 \group_begin:
2456 \char_set_catcode_other:N \#
2457 \use:n
2458 {
2459     \group_end:
2460     \ExplSyntaxOff
2461     \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2462     { \ctex_patch_failure:N \numberline }
2463     \ExplSyntaxOn
2464     \AtBeginDocument
2465     {
2466         \@ifpackageloaded { tocloft }
2467         {
2468             \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2469             { \ctex_patch_failure:N \numberline }
2470             { } { }
2471         }
2472     }

```

(End definition for \numberline.)

7.8.4 页眉信息的修改

`\ps@headings`

```

2473 <*article>
2474 \ctex_patch_cmd:Nnn \ps@headings { \thesection } { \CTEXthesection }
2475 \if@twoside
2476 \ctex_patch_cmd:Nnn \ps@headings { \thesubsection } { \CTEXthesubsection }
2477 \fi:
2478 </article>
2479 <*book|report>
2480 \ctex_patch_cmd:Nnn \ps@headings
2481 { \@chapapp\ \thechapter.\~\ } { \CTEXthechapter \quad }
2482 \if@twoside
2483 \ctex_patch_cmd:Nnn \ps@headings { \thesection.\~\ } { \CTEXthesection \quad }
2484 \fi:
2485 </book|report>

(End definition for \ps@headings.)

2486 \pagestyle { headings }

2487 \bool_if:NT \l__ctex_fancyhdr_bool
2488 { \RequirePackage { fancyhdr } }

```

`\ps@fancy`

```

2489 \if_cs_exist:N \ps@fancy
2490 <*article>
2491 \ctex_patch_cmd:Nnn \ps@fancy
2492 { \thesection \hspace 1em \relax } { \CTEXthesection \quad }
2493 \ctex_patch_cmd:Nnn \ps@fancy
2494 { \thesubsection \hspace 1em \relax } { \CTEXthesubsection \quad }
2495 </article>
2496 <*book|report>
2497 \ctex_patch_cmd:Nnn \ps@fancy { \@chapapp\ \thechapter.\~\ }
2498 <book> { \if@mainmatter \CTEXthechapter \quad \fi }
2499 <report> { \CTEXthechapter \quad }
2500 \ctex_patch_cmd:Nnn \ps@fancy { \thesection.\~\ } { \CTEXthesection \quad }
2501 </book|report>
2502 \fi:

(End definition for \ps@fancy.)

```

7.8.5 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。

```

2503 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2504 {
2505   \protected@edef \@currentlabel
2506   {
2507     \cs_if_exist:CTF { CTEX@the#1 }
2508     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2509     { \exp_not:o { \@currentlabel } }
2510   }
2511 }

(End definition for \refstepcounter.)

```

`\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。

```

2512 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2513 {
2514   \seq_map_inline:Nn \c__ctex_headings_seq
2515   { \ctex_fix_varioref_label:n { ##1 } }
2516 }

```

(End definition for \ctex_varioref_hook:.)

`_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^AT_EX 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。

```
2517 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
2518 {
2519   \tl_if_empty:cT { p@#1 }
2520   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2521 }
```

(End definition for _ctex_fix_varioref_label:n.)

如果 `varioref` 已经被载入, 则使用它来设置。

```
2522 \if_bool:N \l__ctex_caption_bool
2523 \@ifpackageloaded { varioref }
2524 { \ctex_varioref_hook: }
2525 {
2526   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
2527   \RenewDocumentCommand \refstepcounter { m }
2528   {
2529     \CTEX@save@refstepcounter {#1}
2530     \CTEX@setcurrentlabel@n {#1}
2531   }
2532   \AtBeginDocument
2533   { \@ifpackageloaded { varioref } { \ctex_varioref_hook: } { } }
2534 }
2535 \fi:
```

7.8.6 模拟标准文档类格式

下面使用 C_TE_X 文档类的设置方式, 重新模拟标准文档类直接定义或以 `\@startsection` 设定的章节标题格式。

```
2536 \keys_set:nn { ctex / part }
2537 {
2538   name      = \partname \space ,
2539   number    = \thepart ,
2540   <*article>
2541     format   = \raggedright ,
2542     nameformat = \Large \bfseries ,
2543     aftername = \par \nobreak ,
2544     titleformat = \huge \bfseries ,
2545     beforekip = -4ex ,
2546     afterskip  = 3ex
2547   </article>
2548   <*book|report>
2549     format   = \centering ,
2550     nameformat = \huge \bfseries ,
2551     aftername = \par \vskip 20 \p@ ,
2552     titleformat = \Huge \bfseries ,
2553     pagestyle = plain
2554   </book|report>
2555 }
2556 <*book|report>
2557 \keys_set:nn { ctex / chapter }
2558 {
2559   name      = \chaptername \space ,
2560   number    = \thechapter ,
2561   format    = \raggedright ,
2562   nameformat = \huge \bfseries ,
```

```

2563     aftername    = \par \nobreak \vskip 20 \p@ ,
2564     titleformat = \Huge \bfseries ,
2565     beforeskip  = -50 \p@ ,
2566     afterskip   = 40 \p@ ,
2567     pagestyle   = plain
2568   }
2569 </book|report>

2570 \keys_set:nn { ctex / section }
2571 {
2572     number       = \thesection ,
2573     format       = \Large \bfseries ,
2574     aftername    = \quad ,
2575     beforeskip   = -3.5ex \@plus -1ex \@minus -.2ex ,
2576     afterskip    = 2.3ex \@plus .2ex
2577 }

2578 \keys_set:nn { ctex / subsection }
2579 {
2580     number       = \thesubsection ,
2581     format       = \large \bfseries ,
2582     aftername    = \quad ,
2583     beforeskip   = -3.25ex \@plus -1ex \@minus -.2ex ,
2584     afterskip    = 1.5ex \@plus .2ex
2585 }

2586 \keys_set:nn { ctex / subsubsection }
2587 {
2588     number       = \thesubsubsection ,
2589     format       = \normalsize \bfseries ,
2590     aftername    = \quad ,
2591     beforeskip   = -3.25ex \@plus -1ex \@minus -.2ex ,
2592     afterskip    = 1.5ex \@plus .2ex
2593 }

2594 \keys_set:nn { ctex / paragraph }
2595 {
2596     number       = \theparagraph ,
2597     format       = \normalsize \bfseries ,
2598     aftername    = \quad ,
2599     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
2600     afterskip    = -1em
2601 }

2602 \keys_set:nn { ctex / subparagraph }
2603 {
2604     number       = \thesubparagraph ,
2605     format       = \normalsize \bfseries ,
2606     aftername    = \quad ,
2607     beforeskip   = 3.25ex \@plus 1ex \@minus .2ex ,
2608     afterskip    = -1em ,
2609     indent       = \parindent
2610 }

    处理 sub3section 与 sub4section 的格式。
2611 \int_compare:nNnT \g__ctex_section_depth_flag > \c_two
2612 {
2613     \keys_set:nn { ctex / paragraph }
2614     {
2615         beforeskip = -3.25ex \@plus 1ex \@minus .2ex ,
2616         afterskip  = 1ex \@plus .2ex
2617     }
2618 }
2619 \int_compare:nNnT \g__ctex_section_depth_flag > \c_three
2620 {
2621     \keys_set:nn { ctex / subparagraph }
2622     {
2623         beforeskip = -3.25ex \@plus 1ex \@minus .2ex ,
2624         afterskip  = 1ex \@plus .2ex
2625     }

```

```

2626 }
2627 \int_compare:nNtT \g__ctex_section_depth_flag > \c_two
2628 {
2629   \keys_set:nn { ctex / subparagraph }
2630   { indent = \z@ }
2631 }

```

处理附录的格式。

```

2632 <article>\keys_set:nn { ctex / appendix } { name = { } }

```

7.8.7 汉化默认标题格式

在 cap 选项为 true 时, 设置中文化的标题格式。

```

2633 \if_bool:N \l__ctex_caption_bool
2634 \keys_set:nn { ctex / part }
2635 {
2636   number      = \chinese { part } ,
2637 <*article>
2638   format      = \centering ,
2639   aftername   = \quad ,
2640   titleformat = \Large\bfseries ,
2641   beforeskip  = 4ex
2642 </article>
2643 <*book|report>
2644   titleformat = \huge \bfseries
2645 </book|report>
2646 }
2647 <*book|report>
2648 \keys_set:nn { ctex / chapter }
2649 {
2650   number      = \chinese { chapter } ,
2651   format      = \centering ,
2652   aftername   = \quad ,
2653   titleformat = \huge \bfseries
2654   beforeskip  = 50\p@ ,
2655 }
2656 </book|report>
2657 \keys_set:nn { ctex / section }
2658 {
2659   format      = \Large \bfseries \centering ,
2660   beforeskip  = 3.5ex \@plus 1ex \@minus .2ex
2661 }
2662 \keys_set:nn { ctex / subsection }
2663 { beforeskip = 3.25ex \@plus 1ex \@minus .2ex }
2664 \keys_set:nn { ctex / subsubsection }
2665 { beforeskip = 3.25ex \@plus 1ex \@minus .2ex }
2666 \keys_set:nn { ctex / paragraph }
2667 { beforeskip = 3.25ex \@plus 1ex \@minus .2ex }
2668 \keys_set:nn { ctex / subparagraph }
2669 { beforeskip = 3.25ex \@plus 1ex \@minus .2ex }
2670 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
2671 { \ctex_file_input:n { ctexcap-gbk.cfg } }
2672 { \ctex_file_input:n { ctexcap-utf8.cfg } }
2673 \fi:
2674 </class|heading>
2675 <*style>

```

7.8.8 ctex.sty 的 heading 选项

```

2676 \msg_new:nnn { ctex } { not-standard-class }
2677 {
2678   None~of~the~standard~document~classes~was~loaded.\\
2679   ctex~may~not~work~as~expected.
2680 }
2681 \bool_if:NTF \l__ctex_heading_bool

```



```

2682 {
2683   \clist_map_inline:nn { article , book , report }
2684   {
2685     \@ifclassloaded {#1}
2686     { \clist_map_break:n { \tl_const:Nn \c__ctex_class_tl {#1} } } { }
2687   }
2688   \tl_if_exist:NTF \c__ctex_class_tl
2689   { \ctex_file_input:n { ctex- \c__ctex_class_tl .def } }
2690   {
2691     \msg_warning:nn { ctex } { not-standard-class }
2692     \cs_if_exist:NTF \chapter
2693     {
2694       \cs_if_exist:NF \if@mainmatter
2695       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
2696       \ctex_file_input:n { ctex-book.def }
2697     }
2698     { \ctex_file_input:n { ctex-article.def } }
2699   }
2700 }
2701 {
2702   \bool_if:NT \l__ctex_caption_bool
2703   {
2704     \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
2705     { \ctex_file_input:n { ctexcap-gbk.cfg } }
2706     { \ctex_file_input:n { ctexcap-utf8.cfg } }
2707   }
2708 }
2709 </style>
2710 <*UTF8|GBK>

```

7.8.9 标题配置文件

```

2711 \keys_set:nn { ctex }
2712 {
2713   contentsname = 目录 ,
2714   listfigurename = 插图 ,
2715   listtablename = 表格 ,
2716   figurename = 图 ,
2717   tablename = 表 ,
2718   abstractname = 摘要 ,
2719   indexname = 索引 ,
2720   bibname = 参考文献 ,
2721   appendixname = 附录
2722 }
2723 \keys_if_exist:nnT { ctex / part } { name }
2724 {
2725   \keys_set:nn { ctex / part } { name = { 第 , 部分 } }
2726   \keys_if_exist:nnT { ctex / chapter } { name }
2727   { \keys_set:nn { ctex / chapter } { name = { 第 , 章 } } }
2728 }
2729 </UTF8|GBK>

```

7.9 其它功能

\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。同 `hologo` 宏包的设置类似, `CTeX` 可以用在 `\csname` 和 PDF 书签中。

```

2730 <*class|style>
2731 \NewDocumentCommand \CTeX { }
2732 { \ifincsname CTeX \else: C \TeX \fi: }
2733 \AtBeginDocument
2734 {
2735   \cs_if_exist_use:NT \pdfstringdefDisableCommands
2736   { { \tl_set:Nn \CTeX { CTeX } } }

```

```

2737 }
2738 </class|style>

```

(End definition for \CTeX. This function is documented on page 22.)

captiondelimit 过时选项。

```

2739 <*class|style>
2740 \keys_define:nn { ctex }
2741 {
2742   captiondelimit .code:n =
2743   {
2744     \msg_warning:nnn { ctex } { deprecated-option }
2745     { You~can~load~the~package~`caption'~to~get~its~functionality. }
2746   }
2747 }
2748 </class|style>

```

(End definition for captiondelimit.)

7.9.1 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 2749 <*class|heading>
2750 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
2751 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
2752 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
2753 </class|heading>

```

(End definition for \verse and \quotation.)

```

2754 <*class|style>

```

7.9.2 其它兼容性修改

\end breqn 宏包对 **\end** 作了如下处理,然而这个处理并不保险。

```

\def\@tempa#1\endcsname#2\@nil{\def\latex@end##1{#2}}
\expandafter\@tempa\end{#1}\@nil
\def\end#1{\csname end#1\endcsname \latex@end{#1}}%

```

etoolbox 在 **\end** 定义中的 **\csname** 前加入钩子 **\csuse{@end@#1@hook}**。如果 etoolbox 先于 breqn 被载入(这在使用 ctexart 等文档类时几乎是必然的),**\csuse{@end@#1@hook}** 将会被忽略,即 **\AtEndEnvironment** 失效。如果交换两个宏包的载入顺序,则 etoolbox 会给出警告:**\AfterEndEnvironment** 失效,我们不打算处理这种情况。我们通过一个特殊的环境来完成检查。

```

2755 \newenvironment { __ctex_test_env }
2756 { \bool_gset_false:N \g__ctex_tmp_bool } { }
2757 \AtEndEnvironment { __ctex_test_env }
2758 { \bool_gset_true:N \g__ctex_tmp_bool }
2759 \group_begin:
2760 \char_set_catcode_other:N \#
2761 \cs_new_protected_nopar:Npn \ctex_fix_end_env_hook:
2762 {
2763   \begin { __ctex_test_env } \end { __ctex_test_env }
2764   \bool_if:NF \g__ctex_tmp_bool
2765   {
2766     \ctex_patch_cmd:NnnTF \end { \csname end#1 \endcsname }
2767     {
2768       \csuse { @end@#1@hook }
2769       \csname end#1 \endcsname
2770     } { }
2771     {
2772       \ctex_preto_cmd:NnTF \end { \csuse { @end@#1@hook } }
2773       { } { \ctex_patch_failure:N \end }
2774   }

```

```

2775     }
2776 }
2777 \group_end:
2778 \AtBeginDocument { \ctex_fix_end_env_hook: }

(End definition for \end.)

```

7.10 载入中文字体

`\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。

```

2779 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
2780 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
2781 \msg_new:nnn { ctex } { fontset-unavailable }
2782 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }

(End definition for \ctex_fontset_error:n.)

```

`\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。

```

2783 \cs_new_protected_nopar:Npn \ctex_load_fontset:
2784 {
2785   \tl_if_empty:NTF \g__ctex_fontset_tl
2786   {
2787     \ctex_detected_platform:
2788     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2789   }
2790   {
2791     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
2792     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
2793     {
2794       \use:x
2795       {
2796         \ctex_detected_platform:
2797         \msg_error:nxxx { ctex } { fontset-not-found }
2798         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
2799       }
2800       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2801     }
2802   }
2803 }
2804 \@onlypreamble \ctex_load_fontset:
2805 \msg_new:nnnn { ctex } { fontset-not-found }
2806 {
2807   CTeX~fontset~`#1'~could~not~be~found.\\
2808   Fontset~`#2'~will~be~used~instead.
2809 }
2810 { You~may~run~`mktexlsr'~firstly. }

(End definition for \ctex_load_fontset:.)

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

2811 \keys_define:nn { ctex }
2812 {
2813   fontset .code:n =
2814   {
2815     \str_if_eq:x:nnTF {#1} { none }
2816     { \msg_warning:nnn { ctex } { invalid-value } {#1} }
2817     {
2818       \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
2819       {
2820         \tl_gset:Nx \g__ctex_fontset_tl {#1}
2821         \ctex_load_fontset:
2822       }
2823       {
2824         \msg_error:nxxx { ctex } { fontset-loaded }
2825         { \g__ctex_fontset_tl } {#1}

```

```

2826         }
2827     }
2828 }
2829 }
2830 \msg_new:nnnn { ctex } { fontset-loaded }
2831 {
2832     CTeX~fontset~`#1'~has~been~loaded.
2833     \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
2834 }
2835 { Only~one~fontset~can~be~loaded~in~the~preamble. }
2836 \AtBeginDocument
2837 {
2838     \keys_define:nn { ctex }
2839     { fontset .code:n = \msg_warning:nn { ctex } { invalid-option } }
2840 }

```

(End definition for *fontset*. This function is documented on page 7.)

载入中文字库。

```

2841 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
2842 { \ctex_load_fontset: }

```

7.11 宏包配置文件

7.11.1 ctex.cfg

```

2843 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
2844 </class|style>
2845 <*config>
2846 %%
2847 </config>

```

7.11.2 ctexopts.cfg

```

2848 <*ctexopts>
2849 %%
2850 %% \keys_set:nn { ctex / option } { fontset = windows }
2851 </ctexopts>

```

7.12 字体定义文件

7.12.1 传统定义方式

```

2852 <*c19|c70>
2853 %%
2854 %% Chinese characters
2855 %%
2856 <c19>%% character set: GBK (extension of GB 2312)
2857 <c70>%% character set: Unicode
2858 %% font encoding: Unicode
2859 %%
2860 </c19|c70>
2861 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
2862 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
2863 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
2864 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
2865 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
2866 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}
2867 <*rm>
2868 <*c19>
2869 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
2870 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
2871 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
2872 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
2873 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
2874 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}

```

```

2875 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
2876 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
2877 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
2878 </c19>
2879 <*c70>
2880 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
2881 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
2882 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
2883 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
2884 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
2885 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
2886 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
2887 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
2888 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
2889 </c70>
2890 </rm>
2891 <*sf>
2892 <*c19>
2893 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
2894 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
2895 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
2896 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
2897 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
2898 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
2899 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
2900 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
2901 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
2902 </c19>
2903 <*c70>
2904 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
2905 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
2906 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
2907 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
2908 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
2909 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
2910 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
2911 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
2912 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
2913 </c70>
2914 </sf>
2915 <*tt>
2916 <*c19>
2917 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
2918 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
2919 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
2920 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
2921 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
2922 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
2923 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
2924 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
2925 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
2926 </c19>
2927 <*c70>
2928 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
2929 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
2930 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
2931 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
2932 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
2933 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
2934 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
2935 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
2936 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
2937 </c70>
2938 </tt>
2939 <*fontset>
2940 <*windows>

```

7.12.2 ctex-fontset-windows.def

```

2941 \pdfTeX_if_engine:TF
2942 {
2943   \ctex_zhmap_case:nnn
2944   {
2945     \ctex_punct_set:n { windowns }
2946     \setCJKmainfont
2947     [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
2948     \file_if_exist:nTF { C:/boot.ini }
2949     {
2950       \setCJKsansfont { simhei.ttf }
2951       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
2952     }
2953     {
2954       \file_if_exist:nTF { C:/Windows/Fonts/msyh.ttc }
2955       {
2956         \setCJKsansfont [ BoldFont = msyhbd.ttc ] { msyh.ttc }
2957         \setCJKfamilyfont { zhyahei }
2958         [ BoldFont = msyhbd.ttc ] { msyh.ttc }
2959       }
2960       {
2961         \setCJKsansfont [ BoldFont = msyhbd.ttf ] { msyh.ttf }
2962         \setCJKfamilyfont { zhyahei }
2963         [ BoldFont = msyhbd.ttf ] { msyh.ttf }
2964       }
2965       \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
2966       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
2967     }
2968     \setCJKmonofont { simfang.ttf }
2969     \setCJKfamilyfont { zhkai } { simkai.ttf }
2970     \setCJKfamilyfont { zhfs } { simfang.ttf }
2971     \setCJKfamilyfont { zhsong } { simsun.ttc }
2972     \setCJKfamilyfont { zhhei } { simhei.ttf }
2973     \setCJKfamilyfont { zhli } { simli.ttf }
2974     \setCJKfamilyfont { zhyou } { simyou.ttf }
2975     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
2976     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
2977     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
2978     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
2979   }
2980   {
2981     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwinfonts }
2982     \ctex_punct_set:n { windowns }
2983     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
2984     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
2985     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
2986   }
2987   {
2988     \tl_set:Nn \CJKrmdefault { rm }
2989     \tl_set:Nn \CJKsfdefault { sf }
2990     \tl_set:Nn \CJKttdefault { tt }
2991   }
2992 }
2993 {
2994   \file_if_exist:nTF { C:/boot.ini }
2995   {
2996     \setCJKmainfont
2997     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
2998     \setCJKsansfont { SimHei }
2999     \setCJKmonofont { FangSong_GB2312 }
3000     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3001     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3002   }
3003   {
3004     \setCJKmainfont
3005     [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }

```

```

3006     \setCJKsansfont
3007     [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3008     \setCJKmonofont { FangSong }
3009     \setCJKfamilyfont { zhkai } { KaiTi }
3010     \setCJKfamilyfont { zhfs } { FangSong }
3011   }
3012   \setCJKfamilyfont { zhsong } { SimSun }
3013   \setCJKfamilyfont { zhhei } { SimHei }
3014   \setCJKfamilyfont { zhli } { LiSu }
3015   \setCJKfamilyfont { zhyou } { YouYuan }
3016   \setCJKfamilyfont { zhyahei }
3017   [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3018 }
3019 </windows>
3020 <*adobe>

```

7.12.3 ctex-fontset-adobe.def

\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。

```

3021 \pdftex_if_engine:TF
3022 {
3023   \ctex_if_pdfmode:TF
3024   { \ctex_fontset_error:n { adobe } }
3025   {
3026     \ctex_zhmap_case:nnn
3027     {
3028       \setCJKmainfont
3029       [
3030         cmap = UniGB-UTF16-H ,
3031         BoldFont = AdobeHeitiStd-Regular.otf ,
3032         ItalicFont = AdobeKaitiStd-Regular.otf
3033       ] { AdobeSongStd-Light.otf }
3034       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3035       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3036       \setCJKfamilyfont { zhsong }
3037       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3038       \setCJKfamilyfont { zhhei }
3039       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3040       \setCJKfamilyfont { zhkai }
3041       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3042       \setCJKfamilyfont { zhfs }
3043       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3044       \ctex_punct_set:n { adobe }
3045       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3046       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3047       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3048       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3049       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3050     }
3051     {
3052       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3053       \ctex_punct_set:n { adobe }
3054       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3055       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3056       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3057     }
3058     { \ctex_fontset_error:n { adobe } }
3059   }
3060 }
3061 {
3062   \setCJKmainfont
3063   [
3064     BoldFont = AdobeHeitiStd-Regular ,
3065     ItalicFont = AdobeKaitiStd-Regular
3066   ] { AdobeSongStd-Light }

```

```

3067 \setCJKsansfont { AdobeHeitiStd-Regular}
3068 \setCJKmonofont { AdobeFangsongStd-Regular}
3069 \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3070 \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3071 \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3072 \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3073 }
3074 </adobe>
3075 <*fandol>

```

7.12.4 ctex-fontset-fandol.def

```

3076 \pdfTeX_if_engine:TF
3077 {
3078   \ctex_if_pdfmode:TF
3079   { \ctex_fontset_error:n { fandol } }
3080   {
3081     \ctex_zhmap_case:nnn
3082     {
3083       \setCJKmainfont
3084       [
3085         cmap = UniGB-UTF16-H ,
3086         BoldFont = FandolSong-Bold.otf ,
3087         ItalicFont = FandolKai-Regular.otf
3088       ] { FandolSong-Regular.otf }
3089       \setCJKsansfont
3090       [
3091         cmap = UniGB-UTF16-H ,
3092         BoldFont = FandolHei-Bold.otf
3093       ] { FandolHei-Regular.otf }
3094       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3095       \setCJKfamilyfont { zhsong }
3096       [
3097         cmap = UniGB-UTF16-H ,
3098         BoldFont = FandolSong-Bold.otf
3099       ] { FandolSong-Regular.otf }
3100       \setCJKfamilyfont { zhhei }
3101       [
3102         cmap = UniGB-UTF16-H ,
3103         BoldFont = FandolHei-Bold.otf
3104       ] { FandolHei-Regular.otf }
3105       \setCJKfamilyfont { zhfs }
3106       [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3107       \setCJKfamilyfont { zhkai }
3108       [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3109       \ctex_punct_set:n { fandol }
3110       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3111       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3112       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3113       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3114       \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3115       \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3116     }
3117     {
3118       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
3119       \ctex_punct_set:n { fandol }
3120       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3121       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3122       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3123     }
3124     { \ctex_fontset_error:n { fandol } }
3125   }
3126 }
3127 {
3128   \setCJKmainfont
3129   [ BoldFont = FandolSong-Bold , ItalicFont = FandolKai ] { FandolSong-Regular }

```



```

3130 \setCJKsansfont [ BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3131 \setCJKmonofont { FandolFang }
3132 \setCJKfamilyfont { zhsong } [ BoldFont = FandolSong-Bold ] { FandolSong }
3133 \setCJKfamilyfont { zhhei } [ BoldFont = FandolHei-Bold ] { FandolHei }
3134 \setCJKfamilyfont { zhfs } { FandolFang }
3135 \setCJKfamilyfont { zhkai } { FandolKai }
3136 }
3137 </fandol>
3138 <*mac>

```

7.12.5 ctex-fontset-mac.def

OS X Mavericks (10.9) 预装的主要简体中文字体如下¹³:

```

/Library/Fonts/Baoli.ttc: 报隶-简,Baoli SC:style=Regular
/Library/Fonts/Hannotate.ttc: 手札体-简,Hannotate SC:style=Bold
/Library/Fonts/Hannotate.ttc: 手札体-简,Hannotate SC:style=Regular
/Library/Fonts/Hanzipen.ttc: 翩翩体-简,HanziPen SC:style=Bold
/Library/Fonts/Hanzipen.ttc: 翩翩体-简,HanziPen SC:style=Regular
/Library/Fonts/Hei.ttf: Hei
/Library/Fonts/Hiragino Sans GB W3.otf: 冬青黑体简体中文 W3,Hiragino Sans GB W3
/Library/Fonts/Hiragino Sans GB W6.otf: 冬青黑体简体中文 W6,Hiragino Sans GB W6
/Library/Fonts/Kai.ttf: Kai
/Library/Fonts/Kaiti.ttc: STKaiti
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Black
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Bold
/Library/Fonts/Kaiti.ttc: 楷体-简,Kaiti SC:style=Regular
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Demibold
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Heavy
/Library/Fonts/Lantinghei.ttc: 兰亭黑-简,Lantinghei SC:style=Extralight
/Library/Fonts/Libian.ttc: 隶变-简,Libian SC
/Library/Fonts/Songti.ttc: STSong
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Black
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Bold
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Light
/Library/Fonts/Songti.ttc: 宋体-简,Songti SC:style=Regular
/Library/Fonts/WawaSC-Regular.otf: 娃娃体-简,Wawati SC
/Library/Fonts/WeibeiSC-Bold.otf: 魏碑-简,Weibei SC
/Library/Fonts/Xingkai.ttc: 行楷-简,Xingkai SC:style=Bold
/Library/Fonts/Xingkai.ttc: 行楷-简,Xingkai SC:style=Light
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Bold
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Light
/Library/Fonts/Yuanti.ttc: 圆体-简,Yuanti SC:style=Regular
/Library/Fonts/YuppySC-Regular.otf: 雅痞-简,Yuppy SC
/Library/Fonts/华文仿宋.ttf: STFangsong
/Library/Fonts/华文细黑.ttf: STHeiti:style=Light
/Library/Fonts/华文黑体.ttf: STHeiti:style=Regular
/System/Library/Fonts/STHeiti Light.ttc: 黑体-简,Heiti SC:style=Light
/System/Library/Fonts/STHeiti Medium.ttc: 黑体-简,Heiti SC:style=Medium

```

在 DVIPDFMx 下,可以通过下述方式使用华文宋体和华文楷体:

```

\special{pdf:mapline unisong@Unicode@ unicode :4:Songti.ttc}
\special{pdf:mapline unikai@Unicode@ unicode :4:Kaiti.ttc}

```

而 \pdfmapline 似乎不支持带索引的 ttc 字体,Songti.ttc 默认使用的是 Songti SC Black,Kaiti.ttc 默认使用的是 Kaiti SC Black。华文黑体不能通过这种方式使用:

```

\special{pdf:mapline unihei@Unicode@ unicode \detokenize{华文黑体}.ttf}

```

DVIPDFMx 不能生成 PDF,报下述错误:

```

** WARNING ** UCS-4 TrueType cmap table...
** ERROR ** Unable to read OpenType/TrueType Unicode cmap table.

```

如果将 CMap 改为 UniGB-UTF16-H,错误信息是

```

** WARNING ** No usable TrueType cmap table found for font "华文黑体.ttf".
** WARNING ** CID character collection for this font is set to "Adobe-GB1"
** ERROR ** Cannot continue without this...

```

¹³<http://support.apple.com/kb/HT5944>

在 pdfTeX 下生成的 PDF 只有方框¹⁴。华文细黑和华文仿宋的情况类似。

```

3139 \pdfTeX_if_engine:TF
3140 { \ctex_fontset_error:n { mac } }
3141 {
3142   \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
3143   \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
3144   \setCJKmonofont { STFangsong }
3145   \setCJKfamilyfont { zhsong } { STSong }
3146   \setCJKfamilyfont { zhhei } { STHeiti }
3147   \setCJKfamilyfont { zhfs } { STFangsong }
3148   \setCJKfamilyfont { zhkai } { STKaiti }
3149 }
3150 

```

7.12.6 ctex-fontset-founder.def

```

3152 \pdfTeX_if_engine:TF
3153 {
3154   \ctex_zhmap_case:nnn
3155   {
3156     \setCJKmainfont
3157       [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
3158     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
3159     \setCJKmonofont { FZFSK.TTF }
3160     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
3161     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
3162     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
3163     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
3164     \setCJKfamilyfont { zhli } { FZLSK.TTF }
3165     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
3166     \ctex_punct_set:n { founder }
3167     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3168     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
3169     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3170     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3171     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3172     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
3173     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
3174   }
3175   {
3176     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
3177     \ctex_punct_set:n { founder }
3178     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3179     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3180     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3181   }
3182   { \ctex_fontset_error:n { founder } }
3183 }
3184 {
3185   \setCJKmainfont
3186     [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }

```

在 WPS For Linux 附带的 5.00 版和家庭版 5.20 版的“方正细黑一_GBK”的字体名称是 FZXiHeiI-Z08。但在网上发现不少文档和资料都是 FZXiHeiI-Z08，而在官方资料《2013 方正字库字体样张》中对应的英文名字是 FZXiHeiI。可以用 Postscript 名字 FZXH1K--GBK1-0 来统一。经测试时发现 (WPS 中的字体)，XeTeX 找该字体时会出现明显的延迟，而用字体文件名 FZXH1K.TTF 又可能会出现大小写问题，遂采用汉字名称。由于 luaotfload 不支持汉字名称，故使用 Postscript 名字，速度不受影响。

```

3187   \setCJKsansfont [ BoldFont = FZHei-B01 ]
3188   { \xetex_if_engine:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
3189   \setCJKmonofont { FZFangSong-Z02 }

```

¹⁴<http://www.newsmth.net/bbscon.php?bid=460&id=312640>

```

3190 \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
3191 \setCJKfamilyfont { zhhei } { FZHei-B01 }
3192 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
3193 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
3194 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
3195 \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
3196 }
3197 </founder>
3198 <*ubuntu>

```

7.12.7 ctex-fontset-ubuntu.def

以下根据 Ubuntu 12.04 的中文字体情况设置。CMap 不清楚应该是什么,指定为 UniGB-UTF16-H 还是有警告:

**** WARNING **** UCS-4 TrueType cmap table...

需要注意的是 uming.ttc 和 ukai.ttc 看起来像有四种字形的样子,但其实只有“令”和“骨”这区区两个字有新字形,其余都取旧字形¹⁵。

```

3199 \pdfTeX_if_engine:TF
3200 {
3201   \ctex_zhmap_case:nnn
3202   {
3203     \setCJKmainfont
3204     [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
3205     \setCJKsansfont { wqy-zenhei.ttc }
3206     \setCJKmonofont { uming.ttc }
3207     \setCJKfamilyfont { zhsong } { uming.ttc }
3208     \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
3209     \setCJKfamilyfont { zhkai } { ukai.ttc }
3210     \ctex_punct_set:n { ubuntu }
3211     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3212     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3213     \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
3214     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3215     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3216   }
3217   {
3218     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
3219     \ctex_punct_set:n { ubuntu }
3220     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3221     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3222     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3223   }
3224   { \ctex_fontset_error:n { ubuntu } }
3225 }
3226 {
3227   \setCJKmainfont [ ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
3228   \setCJKsansfont { WenQuanYi~Zen~Hei }
3229   \setCJKmonofont { AR~PL~UMing~CN }
3230   \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
3231   \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }
3232   \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
3233 }
3234 </ubuntu>

```

7.12.8 中文字体命令

```

3235 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
3236 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
3237 <!ubuntu> \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
3238 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
3239 <*windows> \founder
3240 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }

```

¹⁵<http://www.freedesktop.org/wiki/Software/CJKUnifonts/Download/>

```

3241 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
3242 </windows|founder>
3243 <windows>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
3244 </fontset>
3245 <*zhmap>

```

7.12.9 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

3246 \begingroup\catcode61\catcode48\catcode32=10\relax%
3247 \catcode 35=6 % #
3248 \catcode123=1 % {
3249 \catcode125=2 % }
3250 \toks0{\endlinechar=\the\endlinechar\relax}%
3251 \toks2{\endlinechar=13 }%
3252 \def\x#1 #2 {%
3253   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
3254   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
3255 \x 13 5 % carriage return
3256 \x 32 10 % space
3257 \x 35 6 % #
3258 \x 40 12 % (
3259 \x 41 12 % )
3260 \x 45 12 % -
3261 \x 46 12 % .
3262 \x 47 12 % /
3263 \x 58 12 % :
3264 \x 60 12 % <
3265 \x 61 12 % =
3266 \x 64 11 % @
3267 \x 91 12 % [
3268 \x 93 12 % ]
3269 \x 123 1 % {
3270 \x 125 2 % }
3271 \edef\x#1{\endgroup%
3272   \edef\noexpand#1{%
3273     \the\toks0 %
3274     \let\noexpand\noexpand\noexpand#1%
3275     \noexpand\noexpand\noexpand\undefined%
3276     \noexpand\noexpand\noexpand\endinput}%
3277     \the\toks2}%
3278 \expandafter\x\csname ctex@zhmap@endinput\endcsname
3279 \input ifpdf.sty\relax

```

提供非 L^AT_EX 格式下的 \ProvidesFile。

```

3280 \begingroup
3281 \expandafter\ifx\csname ProvidesFile\endcsname\relax
3282   \long\def\x#1\ProvidesFile#2[#3]{%
3283     #1%
3284     \immediate\write-1{File: #2 #3}%
3285     \expandafter\xdef\csname ver@#2\endcsname{#3}}
3286 \expandafter\x%
3287 \fi
3288 \endgroup

```

7.12.9.1 zhwinfonts.tex

```

3289 <*windows>
3290 \ProvidesFile{zhwinfonts.tex}%
3291 [2014/06/03 v2.0 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
3292
3293 \ifpdf
3294   \pdfmapline{=gbk@UGBK@ <simsun.ttc>
3295   \pdfmapline{=gbksong@UGBK@ <simsun.ttc>
3296   \pdfmapline{=gbkkai@UGBK@ <simkai.ttf>
3297   \pdfmapline{=gbkhei@UGBK@ <simhei.ttf>

```

```

3298 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
3299 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
3300 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
3301
3302 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
3303 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
3304 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
3305 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
3306 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
3307 \pdfmapline{=unili@Unicode@ <simli.ttf}
3308 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
3309
3310 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
3311 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
3312 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
3313 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
3314 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
3315 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
3316
3317 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
3318 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
3319 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
3320 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
3321 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
3322 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
3323
3324 \else
3325 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
3326 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
3327 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
3328 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
3329 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
3330 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
3331 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
3332
3333 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
3334 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
3335 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
3336 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
3337 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
3338 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
3339 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
3340
3341 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
3342 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
3343 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
3344 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
3345 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
3346 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}
3347
3348 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
3349 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
3350 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
3351 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
3352 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
3353 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
3354
3355 </windows>

```

7.12.9.2 zhadobefonts.tex

```

3356 (*adobe)
3357 \ProvidesFile{zhadobefonts.tex}%
3358 [2014/06/03 v2.0 Adobe font map loader for DVIPDFMx (CTEX)]
3359
3360 \ifpdf
3361 %% pdfTeX does not support OTF fonts
3362

```

```

3363 \else
3364 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3365 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3366 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3367 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3368 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3369
3370 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3371 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3372 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3373 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3374 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3375
3376 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3377 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3378 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3379 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3380
3381 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3382 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3383 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3384 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3385
3386 </adobe>

```

7.12.9.3 zhfandolfonts.tex

```

3387 <*fandol>
3388 \ProvidesFile{zhfandolfonts.tex}%
3389 [2014/06/03 v2.0 Fandol font map loader for DVIPDFMx (CTEX)]
3390
3391 \ifpdf
3392 %% pdfTeX does not support OTF fonts
3393
3394 \else
3395 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3396 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3397 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
3398 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
3399 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
3400
3401 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3402 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3403 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
3404 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
3405 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
3406
3407 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3408 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3409 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3410 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3411
3412 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3413 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3414 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3415 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3416
3417 </fandol>

```

7.12.9.4 zhfounderfonts.tex

```

3418 <*founder>
3419 \ProvidesFile{zhfounderfonts.tex}%
3420 [2014/06/03 v2.0 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
3421
3422 \ifpdf
3423 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF>}
3424 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF>}
3425 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF>}

```

```

3426 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
3427 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
3428 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
3429 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
3430
3431 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
3432 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
3433 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
3434 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
3435 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
3436 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
3437 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
3438
3439 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
3440 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
3441 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
3442 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
3443 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
3444 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
3445
3446 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
3447 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
3448 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
3449 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
3450 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
3451 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
3452
3453 \else
3454 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
3455 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
3456 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
3457 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
3458 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
3459 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
3460 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
3461
3462 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
3463 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
3464 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
3465 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
3466 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
3467 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
3468 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
3469
3470 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
3471 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
3472 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
3473 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
3474 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
3475 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
3476
3477 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
3478 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
3479 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
3480 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
3481 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
3482 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
3483
3484 </founder>

```

7.12.9.5 zhubuntufonts.tex

```

3485 <*ubuntu>
3486 \ProvidesFile{zhubuntufonts.tex}%
3487 [2014/06/03 v2.0 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
3488
3489 \ifpdf
3490 \pdfmapline{=gbk@UGBK@ <uming.ttc}

```

```

3491 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
3492 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
3493 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
3494 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
3495 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
3496
3497 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
3498 \pdfmapline{=unisong@Unicode@ <uming.ttc}
3499 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
3500 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
3501 \pdfmapline{=unifs@Unicode@ <uming.ttc}
3502 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
3503
3504 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
3505 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
3506 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
3507 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
3508 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
3509
3510 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
3511 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
3512 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
3513 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
3514 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
3515
3516 \else
3517 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
3518 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
3519 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
3520 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
3521 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
3522
3523 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
3524 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
3525 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
3526 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
3527 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
3528
3529 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
3530 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
3531 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
3532 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
3533
3534 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
3535 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
3536 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
3537 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
3538
3539 </ubuntu>
3540 \fi
3541
3542 \ctex@zhmap@endinput
3543 </zhmap>

```

7.12.10 制作 spa 文件

我们通过 X_YTEX 的 \XeTeXglyphbounds 取得字体中标点符号的边界信息,为 CJKpunct 宏包制作 spa。

```

3544 <*spa>
3545 <*macro>
3546 \input expl3-generic %
3547 \ExplSyntaxOn
3548 \xetex_if_engine:F
3549 {
3550   \msg_new:nn { ctex } { xetex }
3551   { XeTeX~is~required~to~compile~this~document! }

```


下面是 C_T_EX 定义的一些字体。

```

3603 <*make>
3604 \input ctexspamacro %
3605
3606 \MAKESPA {ctexpunct.spa}
3607 {
3608     {adobezhhsong}      {AdobeSongStd-Light} ,
3609     {adobezhhei}       {AdobeHeitiStd-Regular} ,
3610     {adobezhkai}       {AdobeKaitiStd-Regular} ,
3611     {adobezhfs}        {AdobeFangsongStd-Regular} ,
3612     {fandolzhhsong}    {FandolSong} ,
3613     {fandolzhsongb}    {FandolSong-Bold} ,
3614     {fandolzhhei}      {FandolHei} ,
3615     {fandolzhheib}     {FandolHei-Bold} ,
3616     {fandolzhkai}      {FandolKai} ,
3617     {fandolzhfs}       {FandolFang} ,
3618     {founderzhhsong}   {FZShuSong-Z01} ,
3619     {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
3620     {founderzhhei}     {FZHei-B01} ,
3621     {founderzhheil}    {FZXiHeiI-Z08} ,
3622     {founderzhkai}     {FZKai-Z03} ,
3623     {founderzhfs}      {FZFangSong-Z02} ,
3624     {founderzhli}      {FZLiShu-S01} ,
3625     {founderzhyou}     {FZXiYuan-M01} ,
3626     {founderzhyoub}    {FZZhunYuan-M02} ,
3627     {ubuntuzhsong}     {AR PL UMinG CN} ,
3628     {ubuntuzhhei}      {WenQuanYi Zen Hei} ,
3629     {ubuntuzhkai}      {AR PL UKai CN} ,
3630     {windowszhsong}    {SimSun} ,
3631     {windowszhhei}     {SimHei} ,
3632     {windowszhkai}     {KaiTi} ,
3633     {windowszhfs}      {FangSong} ,
3634     {windowszhli}      {LiSu} ,
3635     {windowszhyou}     {YouYuan} ,
3636     {windowszhyahui}   {Microsoft YaHei} ,
3637     {windowszhyahuib}  {Microsoft YaHei Bold}
3638 }
3639
3640 \primitive\end
3641 </make>
3642 </spa>

```

8 版本历史

v2.0

General: \CTEXindent, \CTEXnoindent 是过时命令。... 68
 \CTEXsetup, \CTEXoptions 是过时命令。... 13, 33
 \CTEXunderdot, \CTEXunderline, \CTEXunderdblline,
 \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令;
 CTEXfilltwosides 是过时环境。... 59
 captiondelimiter 是过时选项。... 82
 indent, noindent 是过时选项。... 29
 nocap 是过时选项。... 31
 nofonts, adobefonts, winfonts 是过时选项。... 29
 nopunct 是过时选项。... 30
 nospace 是过时选项。... 31
 nozhmap 是过时选项。... 30
 punct 选项可以设置标点格式。... 30
 ctex.sty 新增 heading 选项。... 31
 标题设置新增 pagestyle 选项。... 71

调整 \footnotesep 的大小, 以适合行距的变化。... 68
 关闭 fntef 选项下的彩色设置。... 59
 将标题汉化功能加入 ctex.sty。... 69
 将中文字号功能提取到可以独立使用的 ctexsize。... 62
 删除 c19gsn.fd 和 c19gkai.fd。... 1
 通过 luatexja 宏包支持 Lua \TeX 。... 37
 新增 autoindent 选项。... 29
 新增 fontset 选项。... 29
 新增 linespread 选项。... 29
 新增 zhmCJK 支持选项。... 30
 新增统一设置接口 \ctexset。... 33
 应用 \LaTeX 3 重新整理代码。... 1
 中文字号不再采用近似值。... 62
 自动检测操作系统, 载入对应的字体配置。... 57
 \CJK@surr: 解决与 \nouppercase 的冲突。... 35
 \end: 解决 etoolbox 与 breqn 关于 \end 的冲突。... 82

9 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。

Symbols

\# 105, 2456, 2760
 .../format+ 15
 .../nameformat+ 15
 .../numberformat+ 16
 .../titleformat+ 16
 \ / 49, 798, 1158, 1159
 \< 48
 \ \ 13, 14, 33, 49, 126, 161, 163,
 165, 367, 557, 953, 1035, 1729, 1730, 2079, 2678, 2807, 2833
 \{ 48, 964, 965
 \} 964, 965
 10pt 5, 32
 11pt 5, 32
 12pt 5, 32
 \sq 2481, 2483, 2497, 2500

A

\A 50, 51
 \abovedisplayshortskip . 1779, 1787, 1798, 1817, 1825, 1836
 \abovedisplayskip 1778, 1781,
 1786, 1793, 1797, 1804, 1816, 1819, 1824, 1831, 1835, 1842
 \abstractname 12, 2070
 abstractname 12
 \addCJKfontfeature 53, 1035, 1058
 \addCJKfontfeatures 47, 1053, 1058
 \addcontentsline 2206,
 2207, 2209, 2231, 2232, 2234, 2276, 2280, 2283, 2358, 2372
 \addtocontents 2286, 2287

\addvspace 2192, 2199, 2286, 2287
 adobefonts 22
 \AfterEndEnvironment 82
 \AfterPreamble 628, 633
 AlternateFont 21, 49
 \appendix 76, 2431, 2432
 appendix/name 18
 appendix/number 19
 \appendixname 13, 18, 18, 18, 2072, 2426
 appendixname 13
 \apptocmd 101
 \AtBeginDocument
 35, 425, 483, 1453, 2464, 2532, 2733, 2778, 2836
 \AtBeginDvi 424
 \AtBeginShipoutFirst 426, 1555
 \AtBeginUTFCommand 599
 \AtEndEnvironment 82, 2757
 \AtEndOfClass 28, 148
 \AtEndOfPackage 28, 149
 \AtEndPreamble 396, 484, 1486, 1551
 \AtEndUTFCommand 605
 autoindent 8, 11, 29, 65

B

\baselineskip 8, 11, 55, 66, 66
 \baselinestretch 1942
 \begin 2763
 \begingroup 2350, 2385, 3246, 3280
 \belowdisplayshortskip . 1780, 1788, 1799, 1818, 1826, 1837
 \belowdisplayskip 1781, 1793, 1804, 1819, 1831, 1842
 \bfdefault 816, 822, 1140, 1147

- \bfseries 2215, 2218, 2241, 2244, 2253,
2265, 2308, 2313, 2331, 2542, 2544, 2550, 2552, 2562,
2564, 2573, 2581, 2589, 2597, 2605, 2640, 2644, 2653, 2659
- \bibname 13, 2074, 2079, 2082, 2083
- bibname 13
- bool commands:
 - \bool_gset_false:N 249, 254, 2756
 - \bool_gset_true:N 244, 2758
 - \bool_if:NF 2764
 - \bool_if:NT 388, 993, 1969, 2028, 2061, 2487, 2702
 - \bool_if:NTF 2681
 - \bool_if:nTF 57
 - \bool_new:N 25, 146, 1030
 - \bool_set_false:N 201, 1005, 1900
 - \bool_set_true:N 190, 1023, 1896
- box commands:
 - \box_ht:N 2015
 - \box_new:N 28
 - \box_wd:N 1624, 2452
- C**
- c5size 5, 29
- cap 7, 22, 31
- \caption 23
- captiondelimiter 23, 82
- \catcode .. 34, 34, 92, 770, 3246, 3247, 3248, 3249, 3253, 3254
- CCT 22
- CCTfont 22
- \ccwd 9, 9, 9, 9, 11, 11,
23, 60, 60, 60, 61, 65, 66, 66, 67, 1624, 1628, 1630, 1660,
1661, 1925, 1931, 1950, 1951, 1958, 1972, 1985, 1986,
1987, 1990, 1992, 1993, 2004, 2019, 2020, 2750, 2751, 2752
- \centering 2238, 2263, 2549, 2638, 2651, 2659
- \chapter 6, 13, 18, 18, 19, 72, 2692
- chapter/aftername 16
- chapter/afterskip 17
- chapter/beforeskip 16
- chapter/format 15
- chapter/intdent 18
- chapter/name 13
- chapter/nameformat 15
- chapter/number 14
- chapter/numberformat 16
- chapter/pagestyle 18
- chapter/titleformat 16
- \chaptermark 19, 2285
- \chaptername 13, 2559
- \char 38
- char commands:
 - \char_set_catcode:nn 455, 456, 457
 - \char_set_catcode_active:N 1158
 - \char_set_catcode_letter:n 450
 - \char_set_catcode_math_superscript:n 451
 - \char_set_catcode_other:N 51, 105, 135, 136, 2456, 2760
 - \char_set_catcode_other:n 449, 608
 - \char_set_lccode:nn 48, 49, 50, 612, 1159
 - \char_value_catcode:n 455, 456, 457
- CharRange 21, 49
- \CheckCommand 61
- \Chinese 2033
- \chinese 11, 11, 11, 68, 2032, 2033, 2636, 2650
- \CJK 1587
- \CJKbold
2876, 2877, 2887, 2888, 2894, 2895, 2897, 2898, 2900,
2901, 2905, 2906, 2908, 2909, 2911, 2912, 2918, 2919,
2921, 2922, 2924, 2925, 2929, 2930, 2932, 2933, 2935, 2936
- \CJKfamily 44, 45, 434, 436, 438, 440, 890, 912, 1091, 1093,
1095, 1097, 1243, 3235, 3236, 3237, 3238, 3240, 3241, 3243
- \CJKfamilydefault 48,
56, 56, 56, 56, 56, 429, 430, 440, 487, 1086, 1087, 1097,
1104, 1106, 1110, 1117, 1130, 1466, 1470, 1478, 1492, 1494
- \CJKfilltwosides 1596
- \CJKfontspec 47, 1048
- \CJKglue 66, 66, 68, 1623, 1634, 1648, 1657
- \CJKnormal 2869, 2870, 2871, 2872, 2873, 2874, 2875,
2880, 2881, 2882, 2883, 2884, 2885, 2886, 2893, 2896,
2899, 2904, 2907, 2910, 2917, 2920, 2923, 2928, 2931, 2934
- \CJKnospace 8
- \CJKpunctmapfamily
526, 527, 528, 529, 530, 531, 535, 536, 537, 538, 539, 540
- \CJKrmdefault ... 421, 430, 431, 434, 960, 971, 1061, 1087,
1088, 1091, 1106, 1109, 1110, 1474, 2975, 2977, 2978,
2983, 2984, 2985, 2988, 3045, 3048, 3049, 3054, 3055,
3056, 3110, 3113, 3114, 3120, 3121, 3122, 3167, 3170,
3171, 3178, 3179, 3180, 3211, 3214, 3215, 3220, 3221, 3222
- \CJKsfdefault
422, 432, 436, 961, 972, 1067, 1089, 1093, 1475,
2951, 2965, 2966, 2989, 3046, 3111, 3115, 3168, 3172, 3212
- \CJKspace 8
- \CJKsymbol 61
- \CJKtilde 488
- \CJKttdefault 423, 433, 438, 962, 973,
1072, 1090, 1095, 1476, 2976, 2990, 3047, 3112, 3169, 3213
- \CJKunderdotbasesep 23
- \CJKunderline 23
- clearalternatefont 21, 53
- clist commands:
 - \clist_clear:N 837, 3577
 - \clist_concat:NNN 859
 - \clist_const:Nn 506
 - \clist_gput_right:Nn 321, 340, 342
 - \clist_gput_right:No 331
 - \clist_gset:Nn 1078
 - \clist_if_empty:NT 1165, 1421
 - \clist_if_empty:NTF 1181
 - \clist_map_break:n 2686
 - \clist_map_function:nN 1372, 1386
 - \clist_map_function:xN 1338, 1340
 - \clist_map_inline:Nn 496, 1274
 - \clist_map_inline:nn
310, 518, 1379, 1576, 1606, 1747, 1867, 2402, 2683, 3566
 - \clist_new:N 151, 856, 1079, 1216, 3595

- \clist_put_left:Nx 860
- \clist_put_right:Nn 1022
- \clist_put_right:Nx 1262, 3589
- \clist_set:Nn 1199, 1200
- \clist_set:No 841
- \contentsname 12, 12, 2065
- contentsname 12
- \CS 40, 40
- cs commands:
 - \cs:w 2520
 - \cs_end: 2520
 - \cs_generate_variant:Nn 98, 166, 924, 938, 1011, 1031, 1032, 1195, 1232, 1334, 1372, 1380
 - \cs_gset_eq:cc 998, 1001
 - \cs_gset_eq:NN 115, 169, 170, 245, 250, 255, 299, 301, 316, 320, 671, 675
 - \cs_gset_nopar:Npx 672
 - \cs_gset_protected_nopar:cpx 886, 2406
 - \cs_gset_protected_nopar:Npn 616, 2432
 - \cs_gset_protected_nopar:Npx 710, 1236, 1330
 - \cs_if_exist:cF 666, 708, 746
 - \cs_if_exist:CTF 2507
 - \cs_if_exist:Nf 390, 678, 1235, 1327, 1681, 2694
 - \cs_if_exist:NTF 2692
 - \cs_if_exist_p:c 57
 - \cs_if_exist_use:c 908, 909
 - \cs_if_exist_use:cF 649
 - \cs_if_exist_use:CT 1349
 - \cs_if_exist_use:CTF 930
 - \cs_if_exist_use:NT 426, 1566, 2735
 - \cs_if_free:cF 498, 513, 995
 - \cs_if_free:CTF 1139
 - \cs_if_free:Nf 630, 1542, 1917
 - \cs_if_free:NTF 1903
 - \cs_new:Npn 96, 129, 140, 833, 911
 - \cs_new_eq:cc 1615
 - \cs_new_eq:cN 359, 360, 361, 372, 562, 1601
 - \cs_new_eq:Nc 1668
 - \cs_new_eq:NN 36, 38, 40, 145, 147, 155, 492, 687, 712, 1058, 1064, 1488, 1602, 1603, 1657, 1667, 2032, 2033, 2034, 2400, 2431, 2526, 2695, 3571
 - \cs_new_nopar:Npn 727, 903, 956, 967, 3596
 - \cs_new_protected:Npn 45, 55, 68, 99, 100, 101, 102, 108, 122, 392, 609, 834, 853, 1163, 1251, 1265, 1416, 1662, 1767, 1773, 2165, 2446, 3572
 - \cs_new_protected_nopar:cpx 1582
 - \cs_new_protected_nopar:Npn 42, 148, 149, 419, 445, 462, 494, 511, 516, 524, 533, 647, 656, 669, 674, 676, 688, 702, 729, 744, 757, 811, 857, 865, 884, 904, 914, 939, 977, 989, 991, 1013, 1015, 1100, 1125, 1134, 1179, 1190, 1217, 1226, 1233, 1239, 1259, 1272, 1291, 1305, 1312, 1324, 1344, 1360, 1378, 1381, 1383, 1396, 1408, 1442, 1458, 1464, 1496, 1533, 1592, 1598, 1620, 1631, 1659, 1720, 1735, 1852, 1857, 1904, 1932, 1967, 1977, 1982, 2102, 2128, 2398, 2503, 2512, 2517, 2761, 2779, 2783, 3563, 3587
- \cs_new_protected_nopar:Npx 1044
- \cs_set_eq:cc 500, 514
- \cs_set_eq:cN 1580, 1675
- \cs_set_eq:Nc 2399
- \cs_set_eq:NN 117, 441, 460, 480, 491, 631, 681, 682, 831, 890, 1098, 1235, 1243, 1253, 1254, 1255, 1327, 1328, 1329, 1469, 1702, 1705, 1708, 2039, 2044, 2047, 2052
- \cs_set_nopar:cpx 1616
- \cs_set_protected_nopar:Npn 1634, 1774
- \cs_set_protected_nopar:Npx 1664
- \cs_to_str:N 46, 1043
- \cs_undefine:c 869, 870, 874, 875
- \cs_undefine:N 118, 1256
- cs4size 5, 29
- \csname 81, 82, 774, 776, 778, 794, 2339, 2340, 2355, 2357, 2360, 2362, 2363, 2370, 2371, 2374, 2376, 2377, 2766, 2769, 3278, 3281, 3285
- \csuse 2768, 2772
- \CTeX 22, 81, 2731, 2736
- ctex commands:
 - \ctex_add_to_selectfont:n 61, 1662, 1687, 1916
 - \ctex_appto_cmd:NnTF 27, 101, 1455
 - \ctex_assign_heading_name:nn 71, 2134, 2162, 2425
 - _ctex_assign_heading_name:nnn 71, 2164, 2165
 - \ctex_at_end:n 28, 148, 149, 277, 1657, 2843
 - \ctex_auto_ignorespaces: 36, 492, 1705
 - \l_ctex_autoindent_bool 179, 1896, 1900, 1969
 - _ctex_calc_bounds:nn 3591, 3592, 3596
 - \l_ctex_caption_bool 289, 2061, 2522, 2633, 2702
 - _ctex_ccglue: 1648, 1657
 - \l_ctex_ccglue_skip 1635, 1639, 1642, 1644, 1652, 1943, 1954, 1963, 1999, 2007
 - \ctex_CJK_input:n 34, 445, 460
 - \c_ctex_class_tl 2686, 2688, 2689
 - \ctex_declare_math_sizes:nnnn 64, 1852, 1886
 - _ctex_def_heading_keys:n 70, 2128, 2176
 - \ctex_default_pt:n 27, 129, 1923
 - _ctex_default_pt:w 131, 140
 - \ctex_detected_platform: 57, 1496, 2787, 2796
 - \l_ctex_encoding_tl 28, 142, 143, 204, 205, 400, 404, 412, 415, 487, 1545, 2030, 2670, 2704
 - \c_ctex_engine_file_tl 1520, 1530
 - \l_ctex_family_default_init_tl 56, 1466, 1487, 1489, 1494
 - _ctex_family_default_wrap:n 1469, 1488, 1491
 - \l_ctex_fancyhdr_bool 305, 2487
 - \ctex_file_input:n 26, 42, 44, 334, 401, 406, 407, 542, 2671, 2672, 2689, 2696, 2698, 2705, 2706, 2788, 2792, 2800, 2843
 - \ctex_fix_end_env_hook: 2761, 2778
 - _ctex_fix_varioref_label:n 78
 - \ctex_fix_varioref_label:n 2515, 2517
 - \l_ctex_fntef_bool 304, 1572
 - \g_ctex_font_size_flag 28, 155, 169, 170, 316, 320, 339, 1775
 - \c_ctex_font_size_prop 62, 1722, 1733, 1739, 1769, 1862

- \c_ctex_font_size_seq 1731, 1734, 1745
- \l_ctex_font_size_tl 1722, 1723, 1769, 1770, 1854, 1855
- \ctex_fontset_error:n
..... 83, 2779, 3024, 3058, 3079, 3124, 3140, 3182, 3224
- \g_ctex_fontset_tl ... 208, 1499, 1526, 1536, 1537,
2785, 2788, 2791, 2792, 2798, 2800, 2818, 2820, 2825, 2841
- __ctex_get_font_sizes:Nn 64, 1854, 1857
- \l_ctex_heading_bool 288, 2681
- \c_ctex_headings_seq 69, 2095, 2096, 2173, 2514
- \l_ctex_hyperref_bool 306, 388
- \ctex_if_ccglue_touched: 1645
- \ctex_if_ccglue_touched:TF 60, 1938
- \ctex_if_ccglue_touched_p: 60
- \ctex_if_macosx:TF 57, 1521, 1522, 1524, 1533
- \ctex_if_pdfmode:TF 25, 38, 40, 1548, 3023, 3078
- \l_ctex_indent_bool 190, 201, 2028
- __ctex_initial_heading:n 70, 2102, 2175
- \l_ctex_line_spread_fp 176, 2012
- \l_ctex_line_stretch_tl 1924, 1925, 1930, 1931, 1935
- \ctex_load_fontset: 83, 2783, 2804, 2821, 2842
- \ctex_load_zhmap:n 428
- \ctex_load_zhmap:nnnn
..... 34, 419, 2981, 3052, 3118, 3176, 3218
- \l_ctex_ltj_add_alterate_bool 993, 1005, 1023, 1030
- \ctex_ltj_add_font_features:n 47, 1013, 1031
- \ctex_ltj_add_font_features:nn 47, 1015, 1032
- \ctex_ltj_add_font_features:x 1055
- \ctex_ltj_add_font_features:xn 1014
- __ctex_ltj_alterate_cs:n
..... 870, 873, 874, 875, 908, 909, 911, 996,
999, 1000, 1002, 1003, 1185, 1222, 1284, 1309, 1349, 1366
- \l_ctex_ltj_alterate_family_tl 1247, 1250
- \l_ctex_ltj_alterate_options_clist
..... 1200, 1201, 1212, 1216
- \l_ctex_ltj_alterate_seq
..... 838, 1172, 1173, 1178, 1184, 1192, 1440
- \l_ctex_ltj_base_CJKfamily_tl
..... 839, 855, 906, 1284, 1309
- \l_ctex_ltj_base_family_tl 53,
871, 901, 907, 1228, 1269, 1328, 1347, 1352, 1353, 1365
- __ctex_ltj_change_encoding: 43, 811, 889, 1242
- \l_ctex_ltj_char_range_clist 837,
1154, 1165, 1181, 1199, 1211, 1223, 1246, 1268, 1419, 1421
- \ctex_ltj_char_range_key:nn 55, 1413, 1416
- __ctex_ltj_char_range_parse_feature:w
..... 55, 1424, 1428
- \g_ctex_ltj_char_range_prop ... 54, 1276, 1387, 1392
- \l_ctex_ltj_char_range_tl
..... 1276, 1279, 1385, 1387, 1389, 1391, 1398
- __ctex_ltj_check_family:n 44, 865, 1183
- \ctex_ltj_clear_alterate_font:n 1338, 1344
- \l_ctex_ltj_current_font_tl
..... 39, 649, 653, 654, 666, 708, 713
- \l_ctex_ltj_current_shape_tl 731, 736, 741, 742
- \ctex_ltj_declare_alterate_shape:nnnnnn
..... 52, 1245, 1265, 1267
- \ctex_ltj_declare_char_range:n 1378, 1380
- __ctex_ltj_declare_char_range:nn 1379, 1381
- \ctex_ltj_declare_char_range:nn 54, 1382, 1383
- \ctex_ltj_declare_char_range:x 1375
- \ctex_ltj_def_char_range_key:n 54, 1388, 1408
- \g_ctex_ltj_default_features_clist . 859, 1078, 1079
- \ctex_ltj_ensure_default_family: ... 48, 1100, 1483
- \ctex_ltj_extract_font: 40, 681, 688
- __ctex_ltj_family_csname:n 869, 886, 903, 930
- \g_ctex_ltj_family_font_name_prop
..... 44, 844, 863, 867, 941, 1017, 1102, 1114
- \g_ctex_ltj_family_font_options_prop
..... 44, 845, 864, 1020
- \ctex_ltj_family_if_exist:nTF 45
- \ctex_ltj_family_if_exist:xN 925
- \ctex_ltj_family_if_exist:xNF 1104
- \ctex_ltj_family_if_exist:xNT 1130
- \ctex_ltj_family_if_exist:xNTF 916, 1109, 1127, 1347
- \g_ctex_ltj_family_int ... 983, 985, 1261, 1262, 1264
- \g_ctex_ltj_family_name_prop
..... 44, 862, 871, 893, 927, 1228
- \l_ctex_ltj_family_tl
... 918, 923, 980, 981, 996, 1000, 1003, 1014, 1341, 1342
- __ctex_ltj_family_unknown_warning:n
..... 46, 921, 939, 1357
- \l_ctex_ltj_font_options_clist 841,
846, 847, 848, 856, 892, 1021, 1022, 1025, 1219, 1220, 1224
- \ctex_ltj_fontspec:nn 46, 977, 990
- __ctex_ltj_fontspec:nnn 991, 1011
- __ctex_ltj_fontspec:xnn 984
- \ctex_ltj_fontspec:xx 989, 1024, 1050
- \g_ctex_ltj_fontspec_family_tl
..... 891, 894, 895, 896, 902, 907, 932
- \g_ctex_ltj_fontspec_prop 979, 1007, 1012
- __ctex_ltj_gset_family_cs:nn 44, 849, 884
- \ctex_ltj_if_alterate_shape_exist:n 722
- \ctex_ltj_if_alterate_shape_exist:nF 704
- \ctex_ltj_if_alterate_shape_exist:nT 691, 731
- \ctex_ltj_if_alterate_shape_exist:nTF 41
- \g_ctex_ltj_jfm_tl 43, 801, 802, 815, 817, 860
- \c_ctex_ltj_math_fam_int 823, 824
- \c_ctex_ltj_math_family_tl
..... 1136, 1138, 1140, 1143, 1147
- \c_ctex_ltj_math_tl
..... 818, 819, 821, 823, 1076, 1127, 1137, 1142, 1146
- __ctex_ltj_msg_def_family_map:n 954, 956
- __ctex_ltj_msg_family_map:n 883, 953, 967
- __ctex_ltj_noembed_wrap:n 831, 833
- __ctex_ltj_patch_external_font:w 41, 694, 727
- \ctex_ltj_pickup_font: 40, 662, 676, 687, 751
- __ctex_ltj_pop_fontname: 665, 674, 753
- __ctex_ltj_push_fontname:n 661, 669, 750
- \ctex_ltj_reset_alterate_font:n 1340, 1360
- \g_ctex_ltj_reset_alterate_prop
..... 876, 1351, 1363, 1371

- \ctex_ltj_save_alternate_family:Nnnn 51, 1210, 1230, 1233
- \ctex_ltj_save_alternate_seq:cn 1185
- \ctex_ltj_save_alternate_seq:Nn 50, 1190, 1195
- \ctex_ltj_save_alternate_seq:Nnnwn 50
- \ctex_ltj_save_alternate_seq:Nnnwnw 1193, 1196
- __ctex_ltj_save_alternate_shape:cn 1283, 1308
- __ctex_ltj_save_alternate_shape:Nnn 53, 1324, 1334
- \ctex_ltj_save_char_range:n 54, 1386, 1393
- \ctex_ltj_save_char_range:nn 1395, 1396
- __ctex_ltj_save_fontname:w 671, 675
- \ctex_ltj_select_alternate_font: 41, 729, 1690
- \ctex_ltj_select_font: 39, 647, 1689
- __ctex_ltj_select_font_aux: 39, 650, 656, 667
- \ctex_ltj_set_alternate_family:coonn 1221
- __ctex_ltj_set_alternate_family:n 897, 904
- \ctex_ltj_set_alternate_family:nn .. 50, 1188, 1217
- \ctex_ltj_set_alternate_family:nnF ... 50, 842, 1179
- \ctex_ltj_set_alternate_family:nnn 51, 1229, 1237, 1239
- \ctex_ltj_set_alternate_family:Nnnnn ... 1226, 1232
- \ctex_ltj_set_alternate_seq:n 49, 1152, 1163
- \ctex_ltj_set_alternate_shape:n ... 1293, 1305, 1316
- \ctex_ltj_set_alternate_shape:nnN .. 53, 1278, 1312
- \ctex_ltj_set_alternate_shape:nnn .. 52, 1281, 1288
- \ctex_ltj_set_alternate_shape:nnnn 1290, 1291
- \ctex_ltj_set_alternate_shape:Nnnnnnn 52, 1268, 1272
- \ctex_ltj_set_family:nnn 44, 834, 854, 1008
- \ctex_ltj_set_family:xxx 853, 1039, 1046, 1061, 1067, 1072, 1076
- \ctex_ltj_subst_font: 40, 682, 702
- \ctex_ltj_swap_cs:NN 51, 1244, 1251
- \ctex_ltj_switch_family:n 914, 924, 1009, 1045
- \ctex_ltj_switch_family:x 913, 981
- __ctex_ltj_tmp:w 1253, 1255, 1256
- \l__ctex_ltj_tmp_seq 1169, 1170, 1173, 1177
- \l__ctex_ltj_tmp_tl 840, 841, 867, 878, 881, 1018, 1026, 1042, 1045, 1046, 1127, 1128, 1130, 1131, 1167, 1168, 1169, 1203, 1205, 1206, 1213, 1423, 1431, 1440
- \l__ctex_ltj_tmpa_tl 1104, 1109
- __ctex_ltj_um_char:Nw 609, 631
- \g__ctex_ltj_unknown_family_seq 943, 945, 950
- __ctex_ltj_update_family_uid:N . 51, 847, 1219, 1259
- \ctex_ltj_update_mathfont: 49, 1122, 1125
- \ctex_ltj_update_mathfont:n 1128, 1131, 1134
- \ctex_ltj_update_xkanjiskip: 55, 1442, 1691
- __ctex_ltj_use_global_options:N 848, 857, 1201, 1220
- \l__ctex_ltj_xkanjiskip_skip 55, 1445, 1447, 1448, 1451, 1452
- \ctex_lua_now_x:n 25, 36, 602, 620, 698, 724, 728, 733, 738, 760, 810, 1307, 1331, 1501
- \c__ctex_macosx_file_tl 1535, 1539
- \ctex_make_spa:nn 97, 3563, 3571
- \ctex_parse_name:NN 26, 45, 99, 100, 101
- \ctex_parse_name:NNn 55, 98
- __ctex_parse_name:nNNNNn 62, 68
- \ctex_parse_name:NNx 46
- __ctex_parse_name:w 81, 87, 96
- __ctex_patch_cmd:Nnn 106, 108
- \ctex_patch_cmd:Nnn 27, 102, 2474, 2476, 2480, 2483, 2491, 2493, 2497, 2500, 2750, 2751, 2752
- \ctex_patch_cmd:NnnTF 27, 99, 113, 2766
- \ctex_patch_failure:N 120, 122, 435, 437, 439, 442, 1092, 1094, 1096, 1099, 1456, 1677, 2462, 2469, 2773
- \ctex_plane_to_utfxvibe:Nn 35, 462, 480
- \ctex_preto_cmd:NnTF 27, 100, 434, 436, 438, 440, 1091, 1093, 1095, 1097, 1672, 2461, 2468, 2772
- \l__ctex_punct_bounds_clist ... 3577, 3583, 3589, 3595
- \c__ctex_punct_family_clist 496, 506
- \l__ctex_punct_font 3575, 3576
- \ctex_punct_map_bfseries:nn 36, 516, 2966, 2978, 2984, 3049, 3055, 3114, 3115, 3121, 3171, 3172, 3173, 3179, 3215, 3221
- \ctex_punct_map_family:nn ... 36, 511, 2951, 2965, 2975, 2976, 2983, 3045, 3046, 3047, 3054, 3110, 3111, 3112, 3120, 3167, 3168, 3169, 3178, 3211, 3212, 3213, 3220
- \ctex_punct_map_itshape:nn 37, 533, 2977, 2985, 3048, 3056, 3113, 3122, 3170, 3180, 3214, 3222
- \ctex_punct_map_series:nnn 520, 521, 524
- \c__ctex_punct_seq 3554, 3555, 3578
- \ctex_punct_set:n 36, 494, 2945, 2982, 3044, 3053, 3109, 3119, 3166, 3177, 3210, 3219
- \l__ctex_punct_tl 264, 482, 483, 549, 804, 805
- __ctex_save_bounds:n 3579, 3587
- __ctex_save_font_size:nn 62, 1735, 1766
- __ctex_save_selectfont: 1668, 1675
- \g__ctex_section_depth_flag 28, 145, 299, 301, 2611, 2619, 2627
- \ctex_set_font_size:Nnn 63, 1767, 1776, 1784, 1795, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1814, 1822, 1833, 1844, 1845, 1846, 1847, 1848, 1849, 1850
- __ctex_set_font_size:nnNn 1770, 1773
- \g__ctex_spa_iow 3565, 3568, 3570, 3580
- \g__ctex_std_options_clist 28, 151, 321, 331, 340, 342, 345, 349, 353
- __ctex_tmp:w 115, 117, 118
- \g__ctex_tmp_bool 25, 25, 2756, 2758, 2764
- \l__ctex_tmp_box 25, 28, 1623, 1624, 2448, 2452
- \l__ctex_tmp_dim 25, 27, 1935, 1936, 1941, 1944, 1950, 1989, 1991, 1993, 2002, 2004
- \l__ctex_tmp_int 25, 26, 465, 467, 468, 470, 474, 476, 1949, 1951, 1952, 1958, 1959, 1991, 1992, 1993, 1996, 1997, 2002, 2004, 2401, 2405, 2409
- \l__ctex_tmp_seq 29
- \l__ctex_tmp_tl 24, 25, 357, 359, 360, 361, 372, 464, 466, 1862, 1863, 2130, 2172, 2178, 2179
- \ctex_update_ccglue: 60, 1631, 1945, 1964, 2008
- \ctex_update_ccwd: 60, 1620, 1939, 2020
- \ctex_update_default_family: 56, 1464, 1486
- \ctex_update_em_unit: 60, 1659, 1934, 1984
- \ctex_update_parindent: ... 67, 1897, 1909, 1967, 2009
- \ctex_update_size: 65, 1903, 1904, 1917

- \ctex_update_stretch: 66, 1908, 1932
 - \ctex_update_ziju: 67, 1911, 1982
 - \ctex_varioref_hook: 77, 2512, 2524, 2533
 - _ctex_write_family:nn 3567, 3572
 - \c_ctex_zero_tl 1906, 1913, 1915
 - \ctex_zhmap_case:nnn
 - 28, 147, 245, 250, 255, 2943, 3026, 3081, 3154, 3201
 - \g_ctex_zhmCJK_bool 28, 146, 244, 249, 254, 411
 - \ctex_zihao:n 62, 1719, 1720
 - \ctex_ziju:n 1976, 1977
 - \l_ctex_ziju_dim 1985, 1986, 1990, 2001, 2003, 2007, 2011
 - \l_ctex_ziju_tl 1906, 1914, 1915, 1979, 1985
 - \ctexbibname 2080, 2089
 - \CTEXcounter 2034
 - \CTEXdigits 12, 12, 12, 68, 2037
 - \CTEXfilltwosides 23, 1592, 1602
 - \CTEXindent 23, 68, 99, 2016, 2018
 - \CTEXnoindent 23, 68, 99, 2022, 2024
 - \CTEXnumber 11, 11, 11, 12, 12, 12, 68, 2035
 - \CTEXoptions 23, 23, 23, 33, 99, 382, 384
 - \ctexset 1, 8, 8, 8, 8, 8,
 - 8, 9, 12, 13, 14, 21, 21, 22, 23, 23, 33, 83, 99, 374, 379, 385
 - \CTEXsetfont 9, 65, 1902, 1926, 1980
 - \CTEXsetup 23, 23, 33, 99, 376, 378
 - \CTEXsout 23, 99
 - \CTEXthechapter 14, 14, 19, 19, 20, 2275, 2278, 2481, 2498, 2499
 - \CTEXtheparagraph 14
 - \CTEXthepart 14, 2207, 2232
 - \CTEXthesection 14, 14, 19, 19, 2474, 2483, 2492, 2500
 - \CTEXthesubparagraph 14
 - \CTEXthesubsection 14, 2476, 2494
 - \CTEXthesubsubsection 14
 - \CTEXunderdblline 23, 99
 - \CTEXunderdot 23, 99
 - \CTEXunderdotbasesep 23
 - \CTEXunderline 23, 23, 99
 - \CTEXunderwave 23, 99
 - \CTEXxout 23, 99
 - \CurrentOption 331
- D**
- \DeclareAlternateKanjiFont 52
 - declarecharrange 21, 54
 - \DeclareFontEncoding 808
 - \DeclareFontFamily 40, 813, 2861, 2862, 2863, 2864, 2865, 2866
 - \DeclareFontShape 40, 814,
 - 816, 2869, 2870, 2871, 2872, 2873, 2874, 2875, 2876,
 - 2877, 2880, 2881, 2882, 2883, 2884, 2885, 2886, 2887,
 - 2888, 2893, 2894, 2895, 2896, 2897, 2898, 2899, 2900,
 - 2901, 2904, 2905, 2906, 2907, 2908, 2909, 2910, 2911,
 - 2912, 2917, 2918, 2919, 2920, 2921, 2922, 2923, 2924,
 - 2925, 2928, 2929, 2930, 2931, 2932, 2933, 2934, 2935, 2936
 - \DeclareFontSubstitution 809
 - \DeclareMathSizes 1855
 - \DeclareRobustCommand 26
 - \DeclareSymbolFont 819, 1137
- \def 767, 773, 786, 792,
 - 1789, 1800, 1827, 1838, 2203, 2227, 2249, 2261, 2270,
 - 2294, 2318, 2337, 2341, 2367, 2382, 2393, 2394, 3252, 3282
 - \defaultCJKfontfeatures 48, 1077, 1085
 - dim commands:
 - \dim_abs:n 2003
 - \dim_add:Nn 1986
 - \dim_compare:nNnF 1971, 1994
 - \dim_compare:nNnTF 1922, 1936, 1987
 - \dim_eval:n 1956, 2002
 - \dim_max:nn 2451
 - \dim_min:nn 2003
 - \dim_new:N 27, 1630, 2011
 - \dim_set:Nn
 - 1624, 1660, 1935, 1941, 1972, 1985, 1989, 2015, 2020, 2449
 - \dim_set_eq:NN 1661
 - \dim_sub:Nn 1993
 - \dim_to_decimal:n 1741, 1742, 1864
 - \dim_to_pt:n 27
 - \dim_to_unit:nn 3599
 - \dim_use:N 132, 2015
 - \dim_zero:N 2026
 - \directlua 585, 586, 768
- E**
- \E 51
 - \edef 780, 3271, 3272
 - \else 789, 794,
 - 796, 2196, 2208, 2233, 2279, 2282, 2290, 2300, 2324,
 - 2344, 2359, 2366, 2373, 2392, 3324, 3363, 3394, 3453, 3516
 - else commands:
 - \else: 39, 414, 725, 1649, 2732
 - \emph 8
 - empty commands:
 - \c_empty_tl 1580
 - \end 82, 82, 82, 99, 2763, 2766, 2772, 2773, 3640
 - \endCJKfilltwosides 1598
 - \endcsname 774,
 - 776, 778, 794, 2339, 2340, 2355, 2357, 2360, 2362, 2363,
 - 2370, 2371, 2374, 2376, 2377, 2766, 2769, 3278, 3281, 3285
 - \endCTEXfilltwosides 1598, 1603
 - \endgroup 2356, 2391, 3271, 3288
 - \endinput 3276
 - \endlinechar 3250, 3251
 - etex commands:
 - \etex_dimexpr:D 132, 1950
 - \etex_glueexpr:D 1942
 - \EverySelectfont 61, 1681, 1684
 - exp commands:
 - \exp_after:wN
 - .. 131, 678, 694, 696, 1528, 1530, 1531, 1723, 1770, 1855
 - \exp_args:cc 2508
 - \exp_args:Nc 71
 - \exp_args:Nnc 78
 - \exp_args:NNf 1431
 - \exp_args:NNNo 2014

- \exp_args:Nno 2520
 - \exp_args:No 466, 3579
 - \exp_args:Nx 277, 484, 546, 1719, 1976
 - \exp_last_unbraced:Nf 81
 - \exp_not:c 63, 713, 1589, 1616, 2108,
2109, 2110, 2114, 2115, 2116, 2117, 2120, 2121, 2124,
2125, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142,
2143, 2145, 2147, 2149, 2151, 2153, 2410, 2411, 2412, 2415
 - \exp_not:N 63, 486,
487, 488, 639, 640, 642, 749, 893, 894, 895, 896, 1210,
1331, 1474, 1475, 1476, 1491, 1739, 2116, 2408, 2415, 2798
 - \exp_not:n 79, 890, 891, 1469, 1584
 - \exp_not:o 641, 643, 892, 1025,
1026, 1211, 1212, 1213, 1237, 1331, 1492, 1665, 2178, 2509
 - \expandafter 794, 795, 797, 3253, 3254, 3278, 3281, 3285, 3286
 - \ExplFileDate 562
 - \ExplSyntaxOff 112, 567, 766, 1671, 2460, 3601
 - \ExplSyntaxOn 596, 800, 1679, 2463, 3547
- F**
- \familydefault 1472
 - fancyhdr 8, 31
 - \fangsong 9, 9, 3237
 - \fi 789, 795, 796, 797, 2190, 2198,
2210, 2217, 2235, 2243, 2281, 2284, 2293, 2302, 2310,
2311, 2326, 2347, 2364, 2378, 2380, 2395, 2498, 3287, 3540
 - fi commands:
 - \fi: 41, 343, 444, 481,
725, 1619, 1649, 1851, 2477, 2484, 2502, 2535, 2673, 2732
 - \figurename 12, 2068
 - figurename 12
 - file commands:
 - \g_file_current_name_tl 561, 1531
 - \file_if_exist:nTF
..... 1516, 1518, 1520, 1535, 2791, 2948, 2954, 2994
 - \file_input:n 43, 424, 426, 453
 - five commands:
 - \c_five 1742
 - fntef 8, 23, 31
 - \font 40, 40, 2861, 2862, 2863, 2864, 2865, 2866
 - font@name commands:
 - \font@name: 783
 - \fontcharwd 60, 60
 - fontset 7, 30, 83
 - \fontsize 1723
 - fontspec commands:
 - \g_fontspec_encoding_tl 812
 - \l_fontspec_family_tl 1270
 - __fontspec_namewrap:n 43, 831
 - \fontspec_set_family:Nnn 891, 1247
 - \footnotesep 68, 68, 68, 99, 2015
 - \footnotesize 68, 1795, 1833, 2014
 - four commands:
 - \c_four 301, 474, 476
 - fp commands:
 - \fp_eval:n 1979, 3598
 - \fp_use:N 1913, 2012
- G**
- GBK 6, 29
 - \gdef 771, 2436, 2437, 2438, 2441, 2442, 2443
 - \globaljfont 40, 696
 - group commands:
 - \group_begin: 47, 59, 104, 111,
134, 601, 607, 611, 658, 680, 706, 748, 836, 888, 1157,
1241, 1326, 1346, 1362, 1468, 2014, 2119, 2455, 2759, 3574
 - \group_end: 54, 70, 110, 116, 120,
139, 605, 615, 627, 663, 684, 719, 752, 851, 898, 1162,
1248, 1332, 1358, 1369, 1480, 2014, 2122, 2459, 2777, 3585
 - \group_insert_after:N 716, 717
- H**
- hbox commands:
 - \hbox_set:Nn 1623, 2448
 - heading 6, 31
 - \heiti 9, 3236
 - \hskip 2352, 2368, 2387, 2393, 2394, 2492, 2494
 - \hspace 2206, 2207, 2231, 2232, 2278
 - \Huge 1812, 1850, 2244, 2265, 2313, 2331, 2552, 2564
 - \huge 1811,
1849, 2218, 2241, 2253, 2308, 2544, 2550, 2562, 2644, 2653
 - hyperref 8, 31
 - \hypersetup 8, 33, 390, 392, 395, 1542, 1544, 1547, 1562, 1566
 - \hyphenchar 2861, 2862, 2863, 2864, 2865, 2866
- I**
- if commands:
 - \if_bool:N 411, 1572, 2522, 2633
 - \if_case:w 339, 1775
 - \if_cs_exist:N 461, 2489
 - \if_meaning:w 1648
 - \ifdim 2194, 2298, 2322, 2349, 2384
 - \ifincsname 2732
 - \ifmmode 792
 - \IfNoValueF 380, 386
 - \IfNoValueTF 1043, 1295, 1400, 1433, 1435, 2168
 - \ifnum ... 2204, 2214, 2229, 2240, 2271, 2306, 2342, 2359, 2373
 - \ifpdf 25, 37, 3293, 3360, 3391, 3422, 3489
 - \ifx 794, 3281
 - \immediate 3284
 - indent 22, 29
 - \indexname 12, 2071
 - indexname 12
 - \input 3279, 3546, 3604
 - int commands:
 - \int_compare:nNnF 1996
 - \int_compare:nNnT 2611, 2619, 2627
 - \int_compare:nNnTF 467, 1952
 - \int_const:Nn 823
 - \int_div_truncate:nn 66, 474
 - \int_eval:n 1296, 1298, 1299, 1401, 1403, 1404
 - \int_from_hex:n 466
 - \int_gincr:N 983, 1261

`\int_incr:N` 2405
`\int_mod:nn` 476
`\int_new:N` 26, 1264
`\int_set:Nn` 454, 465, 1949, 1991
`\int_set_eq:NN` 452
`\int_sub:Nn` 470, 1997
`\int_to_Hex:n` 468, 473, 475
`\int_use:N` 454, 985, 1262, 2409, 3579
`\int_zero:N` 2401
`\interlinepenalty`
 2211, 2237, 2250, 2262, 2312, 2330, 2353, 2354, 2388, 2389
 iow commands:
`\iow_close:N` 3568
`\iow_new:N` 3570
`\iow_now:Nx` 3580
`\iow_open:Nn` 3565
`\itdefault` 40
`\itemsep` 1792, 1803, 1830, 1841

J

`\jfam` 796, 824

K

`\kaishu` 9, 3238
 keys commands:
`\l_keys_choice_tl` 278
`\keys_define:nn` 167, 312, 328, 828, 1150,
 1258, 1335, 1373, 1412, 1696, 1890, 1918, 2040, 2063,
 2083, 2086, 2089, 2178, 2181, 2421, 2423, 2740, 2811, 2838
`\keys_if_exist:nnF` 1410
`\keys_if_exist:nnT` 2723, 2726
`\l_keys_key_tl` 157, 159, 161
`\keys_set:nn` 217, 226, 235, 261, 271, 278,
 285, 295, 375, 380, 386, 1581, 2062, 2536, 2557, 2570,
 2578, 2586, 2594, 2602, 2613, 2621, 2629, 2632, 2634,
 2648, 2657, 2662, 2664, 2666, 2668, 2711, 2725, 2727, 2850
`\keys_set_known:nnN` 840

L

`\labelformat` 78, 78, 2520
`\LARGE` 1810, 1848
`\Large` 1809, 1847, 2215, 2542, 2573, 2640, 2659
`\large` 1808, 1846, 2581
`\leavevmode` 2190
`\leftmargin` 1789, 1800, 1827, 1838
`\leftmargini` 1789, 1800, 1827, 1838
`\leftmark` 19
`\let` 772, 775, 777, 788, 789, 798, 1782, 1820, 2343, 3274
`\linespread` 2012
`linespread` 8, 29
`linestretch` 11, 66
`\linewidth` 66, 66, 1950, 1958, 1990
`\lishu` 9, 9, 3240
`\listfigurename` 12, 2066
`listfigurename` 12
`\listtablename` 12, 2067
`listtablename` 12

`\LoadClass` 346, 350, 354
`\long` 3282
`\lstlistingname` 640, 641
`\lstlistlistingname` 642, 643
`\ltjalchar` 38
`\ltjdefcharrange` 568, 569, 570, 572, 576, 577, 580, 583
`LTJFONTUID` 51
`\ltjgetparameter` 1445, 1628, 1652
`\ltjsetmathletter` 625
`\ltjsetparameter` 55, 584, 590, 1448, 1459, 1642
 lua commands:
`\lua_now_x:n` 25
 luatex commands:
`\luatex_directlua:D` 36
`\luatexja` 41
`\luatexluaescapestring`
 .. 724, 736, 763, 769, 1286, 1301, 1302, 1319, 1320, 1329

M

`\MAKESPA` 3571, 3606
`\markboth` 19, 2220, 2236
`\markright` 19
`\mathbb` 81
`\mathgroup` 796
 max commands:
`\c_max_dim` 1936
`\maxdimen` 11, 66
`\mddefault` 809, 814, 820, 1138, 1143
`\meaning` 26
 minus commands:
`\c_minus_one` 316, 320, 452
 mode commands:
`\mode_if_math:TF` 618
 msg commands:
`\msg_critical:nnn` 2780
`\msg_critical:nnx` 561
`\msg_error:nn` 371
`\msg_error:nnn` 18, 20, 552, 566, 1724, 1771
`\msg_error:nnx` 2056
`\msg_error:nnxx` 2797, 2824
`\msg_fatal:nn` 3552
`\msg_new:nn` 3550
`\msg_new:nnn` 124, 156, 158,
 160, 162, 164, 555, 882, 951, 1033, 1569, 2077, 2676, 2781
`\msg_new:nnnn` 10, 30, 364, 1726, 2058, 2805, 2830
`\msg_warning:nn` 1028, 1575, 1713, 2088, 2691, 2839
`\msg_warning:nncn` 1586
`\msg_warning:nnn` 183,
 195, 212, 221, 230, 259, 269, 283, 293, 946, 2744, 2816
`\msg_warning:nnnn` 166, 378, 384, 1594, 2018, 2024
`\msg_warning:nnx` 123
`\msg_warning:nnxx` 878

N

`\newCJKfontfamily` 47, 1040
`\newcommand` 26

`\NewDocumentCommand` 374, 376, 382, 912, 1038,
 1040, 1048, 1053, 1059, 1065, 1070, 1075, 1077, 1196,
 1288, 1393, 1428, 1718, 1902, 1975, 2016, 2022, 2035,
 2037, 2162, 2731, 3235, 3236, 3237, 3238, 3240, 3241, 3243
`\newenvironment` 2755
`\newfontfeature` 825, 826, 827
`\newrobustcmd` 26
 nil commands:
 `\q_nil` 609
`\nobreak`
 2215, 2221, 2255, 2308, 2313, 2314, 2331, 2332, 2543, 2563
`\nobreakspace` 2215, 2241
`nocap` 22
`\noexpand` 3272, 3274, 3275, 3276
`nofonts` 22
`noindent` 22
`nopunct` 22
`\normalem` 1573
`\normalfont` 440, 441, 442, 1062,
 1068, 1073, 1097, 1098, 1099, 1121, 2212, 2213, 2238,
 2239, 2251, 2252, 2263, 2264, 2304, 2305, 2328, 2329, 2415
`\normalsize` 1776, 1814, 1887, 2013, 2589, 2597, 2605
`nospace` 23
`\nouppercase` 35, 99
`nozhmap` 22
`\numberline` 76,
 2277, 2278, 2360, 2361, 2374, 2375, 2461, 2462, 2468, 2469

O

`\O` 51
 one commands:
 `\c_one` 170, 1289, 1394, 2004, 2163, 3591
 or commands:
 `\or:` 341, 1813

P

`\P` 51, 135
`\pagestyle` 2486
`pagestyle` 71
`\par` 2191, 2215, 2220, 2241, 2244, 2245, 2253, 2254,
 2265, 2266, 2308, 2313, 2314, 2331, 2332, 2543, 2551, 2563
`\paragraph` 6, 6, 6, 16, 17, 28, 28
`paragraph/aftername` 16
`paragraph/afterskip` 17
`paragraph/beforeskip` 16
`paragraph/format` 15
`paragraph/intdent` 18
`paragraph/name` 13
`paragraph/nameformat` 15
`paragraph/number` 14
`paragraph/numberformat` 16
`paragraph/titleformat` 16
`\parindent` 8,
 9, 11, 65, 67, 68, 1971, 1972, 1994, 2019, 2020, 2025,
 2026, 2212, 2213, 2251, 2252, 2304, 2305, 2328, 2329, 2609
`\parsep` 1791, 1792, 1802, 1803, 1829, 1830, 1840, 1841
`\part` 13, 18, 71, 72, 2189

`part/aftername` 16
`part/afterskip` 17
`part/beforeskip` 16
`part/format` 15
`part/intdent` 18
`part/name` 13
`part/nameformat` 15
`part/number` 14
`part/numberformat` 16
`part/pagestyle` 18
`part/titleformat` 16
`\partname` 13, 2215, 2241, 2538
`\PassOptionsToClass` 32, 345, 349, 353
`\PassOptionsToPackage` 3, 393, 412, 2030
`\patchcmd` 26, 99
`\pdfmapline` 87, 89, 3294,
 3295, 3296, 3297, 3298, 3299, 3300, 3302, 3303, 3304,
 3305, 3306, 3307, 3308, 3310, 3311, 3312, 3313, 3314,
 3315, 3317, 3318, 3319, 3320, 3321, 3322, 3423, 3424,
 3425, 3426, 3427, 3428, 3429, 3431, 3432, 3433, 3434,
 3435, 3436, 3437, 3439, 3440, 3441, 3442, 3443, 3444,
 3446, 3447, 3448, 3449, 3450, 3451, 3490, 3491, 3492,
 3493, 3494, 3495, 3497, 3498, 3499, 3500, 3501, 3502,
 3504, 3505, 3506, 3507, 3508, 3510, 3511, 3512, 3513, 3514
`\pdfstringdef` 58
`\pdfstringdefDisableCommands` 2735
 pdftex commands:
 `\pdfTeX_if_engine:TF`
 144, 398, 2941, 3021, 3076, 3139, 3152, 3199
`\pretocmd` 100
 prg commands:
 `\prg_do_nothing:` 1235, 1327, 1603, 1667, 1702
 `\prg_new_conditional:Npnn` 722, 1645
 `\prg_new_protected_conditional:Npnn` 925
 `\prg_return_false:` 725, 935, 1649, 1653
 `\prg_return_true:` 725, 928, 933, 1649, 1653
`\primitive` 3640
`\ProcessKeysOptions` 335
 prop commands:
 `\prop_get:NnN` 1020
 `\prop_get:NnNT` 1228
 `\prop_get:NnNTF` .. 938, 979, 1017, 1276, 1722, 1769, 1862
 `\prop_get:NxNTF` 927
 `\prop_gpop:NnNT` 867, 871, 1363
 `\prop_gput:Nnn` 844, 1007, 1739
 `\prop_gput:Nno` 845, 893, 1351, 1387
 `\prop_gremove:Nn` 876
 `\prop_if_empty:NF` 941, 1102
 `\prop_map_break:n` 1116
 `\prop_map_inline:Nn` 1114
 `\prop_new:N` 862, 863, 864, 1012, 1371, 1392, 1733
`\protect` 75, 76, 2277, 2278, 2286, 2287, 2360, 2361, 2374, 2375
`\protected` 38
`\ProvidesFile` 92, 3282, 3290, 3357, 3388, 3419, 3486
`punct` 7, 22, 30
`\punctstyle` 483

Q

\quad 2339, 2481, 2483, 2492, 2494,
2498, 2499, 2500, 2574, 2582, 2590, 2598, 2606, 2639, 2652
\quotation 82, 2752

R

\raggedright 2212, 2251, 2304, 2328, 2541, 2561
\refname 13, 2073, 2079, 2085, 2086
\refstepcounter .. 77, 77, 2205, 2230, 2273, 2345, 2526, 2527
\relax 35, 65, 771, 792, 795, 796,
2193, 2229, 2240, 2297, 2321, 2346, 2348, 2352, 2368,
2383, 2393, 2394, 2492, 2494, 3246, 3250, 3253, 3279, 3281
\renewcommand 2189
\RenewDocumentCommand 2527
\RequirePackage .. 7, 23, 396, 413, 416, 417, 418, 545, 563,
564, 598, 606, 639, 1549, 1558, 1573, 1574, 2029, 2031, 2488
\RequirePackageWithOptions 4
resetalternatefont 21, 53
\rightmark 19
\rmdefault 1474
\rmfamily 434, 435, 1091, 1092
\roman 11

S

\S 51
scan commands:
 \scan_stop: ... 53, 132, 696, 1328, 1329, 1942, 1950, 3575
\scantokens 26
scriptfont@name commands:
 \scriptfont@name: 782
\scriptsize 1806, 1844
\secdef 2200
\section 13, 13, 18, 19
section/aftername 16
section/afterskip 17
section/beforeskip 16
section/format 15
section/intdent 18
section/name 13
section/nameformat 15
section/number 14
section/numberformat 16
section/titleformat 16
\sectionmark 19, 19
\selectfont 41, 55, 61,
61, 61, 65, 919, 1354, 1367, 1672, 1677, 1693, 1723, 1903
seq commands:
 \seq_clear:N 838
 \seq_concat:NNN 1172
 \seq_gput_right:Nn 945, 1745
 \seq_gset_from_clist:Nn 2096, 3555
 \seq_if_empty:Nf 1184
 \seq_if_in:NnF 943
 \seq_map_inline:Nn 1192, 2173, 2514, 3578
 \seq_new:N 29, 950, 1177, 1178, 1734, 2095, 3554
 \seq_put_right:No 1440
 \seq_set_filter:NNn 1170

 \seq_set_split:NnV 1169
 \seq_use:Nnnn 1731
\setCJKfamilyfont 21, 21, 21, 47, 964, 1038,
2957, 2962, 2969, 2970, 2971, 2972, 2973, 2974, 3000,
3001, 3009, 3010, 3012, 3013, 3014, 3015, 3016, 3036,
3038, 3040, 3042, 3069, 3070, 3071, 3072, 3095, 3100,
3105, 3107, 3132, 3133, 3134, 3135, 3145, 3146, 3147,
3148, 3160, 3161, 3162, 3163, 3164, 3165, 3190, 3191,
3192, 3193, 3194, 3195, 3207, 3208, 3209, 3230, 3231, 3232
\setCJKmainfont 21, 48, 960, 1059, 1064, 1080, 2946, 2996,
3004, 3028, 3062, 3083, 3128, 3142, 3156, 3185, 3203, 3227
\setCJKmathfont 48, 1075, 1083
\setCJKmonofont 48, 962, 1070, 1082, 2968, 2999,
3008, 3035, 3068, 3094, 3131, 3144, 3159, 3189, 3206, 3229
\setCJKromanfont 1064, 1084
\setCJKsansfont 48, 961, 1065, 1081, 2950, 2956, 2961, 2998,
3006, 3034, 3067, 3089, 3130, 3143, 3158, 3187, 3205, 3228
\SetSymbolFont 821, 1142, 1146
\sfdefault 1475
\sffamily 436, 437, 1093, 1094
six commands:
 \c_six 1742
skip commands:
 \skip_horizontal:N 1635, 1639
 \skip_if_eq:nnT 1444
 \skip_if_eq:nnTF 1652
 \skip_new:N 1451, 1644
 \skip_set:Nn ... 1447, 1452, 1628, 1943, 1954, 1999, 2007
 \skip_zero:N 1963
\sldefault 40
\small 1784, 1822
\songti 9, 3235
\space 2274, 2308, 2426, 2538, 2559
space 8, 11, 23, 31, 61
\special .. 6, 58, 1556, 3325, 3326, 3327, 3328, 3329, 3330,
3331, 3333, 3334, 3335, 3336, 3337, 3338, 3339, 3341,
3342, 3343, 3344, 3345, 3346, 3348, 3349, 3350, 3351,
3352, 3353, 3364, 3365, 3366, 3367, 3368, 3370, 3371,
3372, 3373, 3374, 3376, 3377, 3378, 3379, 3381, 3382,
3383, 3384, 3395, 3396, 3397, 3398, 3399, 3401, 3402,
3403, 3404, 3405, 3407, 3408, 3409, 3410, 3412, 3413,
3414, 3415, 3454, 3455, 3456, 3457, 3458, 3459, 3460,
3462, 3463, 3464, 3465, 3466, 3467, 3468, 3470, 3471,
3472, 3473, 3474, 3475, 3477, 3478, 3479, 3480, 3481,
3482, 3517, 3518, 3519, 3520, 3521, 3523, 3524, 3525,
3526, 3527, 3529, 3530, 3531, 3532, 3534, 3535, 3536, 3537
\SplitArgument 71, 1289, 1394, 2163
stop commands:
 \q_stop 82, 87, 96, 132, 140, 1193, 1197, 1424, 1429
str commands:
 \str_case:nnTF 73
 \str_case:onF 1472
 \str_case:onTF 87
 \str_case_x:nnF 958, 969
 \str_if_eq:nnF 2833
 \str_if_eq:onF 2841

<code>\str_if_eq:onTF</code>	415, 1545, 1553, 2670, 2704, 2818
<code>\str_if_eq_x:nnF</code>	482
<code>\str_if_eq_x:nnTF</code>	79, 804, 1106, 2815
<code>\string</code>	774, 776, 778, 787, 795
<code>\strutbox</code>	68, 2015
<code>sub3section</code>	6, 31
<code>sub4section</code>	6, 31
<code>\subparagraph</code>	6, 6, 16, 17, 28
<code>subparagraph/aftername</code>	16
<code>subparagraph/afterskip</code>	17
<code>subparagraph/beforeskip</code>	16
<code>subparagraph/format</code>	15
<code>subparagraph/intdent</code>	18
<code>subparagraph/name</code>	13
<code>subparagraph/nameformat</code>	15
<code>subparagraph/number</code>	14
<code>subparagraph/numberformat</code>	16
<code>subparagraph/titleformat</code>	16
<code>subsection/aftername</code>	16
<code>subsection/afterskip</code>	17
<code>subsection/beforeskip</code>	16
<code>subsection/format</code>	15
<code>subsection/intdent</code>	18
<code>subsection/name</code>	13
<code>subsection/nameformat</code>	15
<code>subsection/number</code>	14
<code>subsection/numberformat</code>	16
<code>subsection/titleformat</code>	16
<code>subsubsection/aftername</code>	16
<code>subsubsection/afterskip</code>	17
<code>subsubsection/beforeskip</code>	16
<code>subsubsection/format</code>	15
<code>subsubsection/intdent</code>	18
<code>subsubsection/name</code>	13
<code>subsubsection/nameformat</code>	15
<code>subsubsection/number</code>	14
<code>subsubsection/numberformat</code>	16
<code>subsubsection/titleformat</code>	16
T	
<code>\T</code>	136
<code>\t</code>	50
<code>\tablename</code>	12, 2069
<code>tablename</code>	12
<code>\TeX</code>	2732
TeX and L^AT_EX 2_ε commands:	
<code>\@italiccorr</code>	798
<code>\@par</code>	2353, 2355, 2388, 2390
<code>\@Alph</code>	2428, 2429
<code>\@EverySelectfont@Init</code>	1674
<code>\@M</code> 2211, 2237, 2250, 2262, 2312, 2330, 2353, 2354, 2388, 2389	
<code>\@afterheading</code>	2224, 2258, 2292
<code>\@afterindentfalse</code>	2195, 2299, 2323
<code>\@afterindenttrue</code>	2197, 2301, 2325
<code>\@chapapp</code>	2274, 2308, 2481, 2497
<code>\@chapter</code>	73, 2270
<code>\@currentlabel</code>	2505, 2509
<code>\@currentx</code>	357
<code>\@currname</code>	357
<code>\@defaultunits</code>	27
<code>\@empty</code>	78, 2343
<code>\@endpart</code>	2246, 2267
<code>\@changefrom</code>	2352, 2387
<code>\@ifclassloaded</code>	2685
<code>\@ifpackagelater</code>	17, 19, 551, 565
<code>\@ifpackageloaded</code>	370, 560, 635, 2466, 2523, 2533
<code>\@ifundefined</code>	2338, 2361, 2375
<code>\@listI</code>	1782, 1820
<code>\@listi</code>	1782, 1789, 1800, 1820, 1827, 1838
<code>\@makechapterhead</code>	73, 2289, 2291, 2294
<code>\@makeschapterhead</code>	74, 2318
<code>\@minus</code>	1778, 1780, 1786, 1788, 1790, 1791, 1797, 1799, 1801, 1802, 1816, 1818, 1824, 1826, 1828, 1829, 1835, 1837, 1839, 1840, 2575, 2583, 2591, 2599, 2607, 2615, 2623, 2660, 2663, 2665, 2667, 2669
<code>\@namedef</code>	3582
<code>\@nil</code>	749
<code>\@onlypreamble</code>	44, 428, 1080, 1081, 1082, 1083, 1084, 1085, 2804
<code>\@part</code>	72, 2200, 2203, 2227
<code>\@pkgextension</code>	359, 360, 361, 372, 562
<code>\@plus</code>	1778, 1779, 1780, 1786, 1787, 1788, 1790, 1791, 1797, 1798, 1799, 1801, 1802, 1816, 1817, 1818, 1824, 1825, 1826, 1828, 1829, 1835, 1836, 1837, 1839, 1840, 2575, 2576, 2583, 2584, 2591, 2592, 2599, 2607, 2615, 2616, 2623, 2624, 2660, 2663, 2665, 2667, 2669
<code>\@popfilename</code>	26, 43
<code>\@pushfilename</code>	26, 43
<code>\@secCNTformat</code>	74, 2337, 2346
<code>\@sect</code>	74, 2341
<code>\@setfontsize</code>	1774
<code>\@spart</code>	72, 2200, 2249, 2261
<code>\@ssect</code>	75, 75, 2382
<code>\@startsection</code>	75, 75, 78, 2408
<code>\@svsec</code>	2343, 2346, 2352, 2369, 2370
<code>\@svsechd</code>	2367, 2393, 2394
<code>\@tempdima</code>	2449, 2451
<code>\@tempskipa</code>	2193, 2194, 2195, 2199, 2297, 2298, 2299, 2303, 2321, 2322, 2323, 2327, 2348, 2349, 2383, 2384
<code>\@topnewpage</code>	2289
<code>\@xsect</code>	2381, 2396
<code>\bfseries</code>	36
<code>\c@chapter</code>	2429
<code>\c@secnumdepth</code>	2204, 2214, 2229, 2240, 2271, 2306, 2342, 2359, 2373
<code>\c@section</code>	2428
<code>\CJK@@ignorespaces</code>	491, 492
<code>\CJK@encoding</code>	43, 43, 44, 52, 655, 659, 743, 807, 808, 810, 812, 813, 814, 816, 820, 822, 1137, 1140, 1142, 1146, 1286, 1301, 1302, 1319, 1320
<code>\CJK@envStart</code>	35, 36, 486

- \CJK@family 39, 39, 650, 652, 655, 660, 707, 743, 916, 1353, 1363, 1365, 1693, 1694
- \CJK@ignorespaces 36, 491, 1702, 1705, 1708
- \CJK@input 34, 460
- \CJK@loadBinding 34
- \CJK@loadEncoding 34
- \CJK@makeActive 34, 443
- \CJK@plane 35, 61, 1681
- \CJK@surr 35, 35, 35, 461, 480
- \CJK@upperReset 36
- \CJKrmdefault 34
- \clearpage 36
- \CTEX@appendixnumber 2427, 2437, 2442
- \CTEX@chapter@afterskip 2316, 2334
- \CTEX@chapter@beforeskip 2297, 2321
- \CTEX@chapter@format 2305, 2329
- \CTEX@chapter@indent 2305, 2329
- \CTEX@chapter@pagestyle 2184, 2295, 2319
- \CTEX@chapter@titleformat 2314, 2332
- \CTEX@chaptername 2309
- \CTEX@part@afterskip 2223, 2257
- \CTEX@part@beforeskip 2193
- \CTEX@part@format 2213, 2239, 2252, 2264
- \CTEX@part@indent 2213, 2252
- \CTEX@part@pagestyle 2183, 2228
- \CTEX@part@titleformat 2219, 2245, 2254, 2266
- \CTEX@partname 2216, 2242
- \CTEX@postappendix 2420, 2438, 2443
- \CTEX@postchapter 2443
- \CTEX@postsection 2438
- \CTEX@preappendix 2419, 2436, 2441
- \CTEX@prechapter 2441
- \CTEX@presection 2436
- \CTEX@save@appendix 2431, 2434
- \CTEX@save@refstepcounter 2526, 2529
- \CTEX@selectfont@hook 61, 1664, 1665, 1667, 1672, 1681, 1684
- \CTEX@set@titleformat@n 75, 2398, 2414
- \CTEX@setcurrentlabel@n 2503, 2530
- \CTEX@thechapter 2442
- \CTEX@thesection 2437
- \CTEX@titleformat@n .. 75, 75, 75, 2390, 2394, 2399, 2400
- \CTEX@toc@width@n 2446, 2461, 2468
- \CTEX@todayold 2039, 2044
- \CTEX@verbatim@font@hook 1455, 1458
- \ctex@zhmap@endinput 3542
- \curr@fontshape 661, 691, 700, 704, 750
- \DeclareFontShape@ 52, 1245
- \define@newfont 40, 683
- \do@subst@correction 40, 40, 682
- \external@font 693, 694, 696
- \extract@font 40, 681
- \f@baselineskip 1942
- \f@encoding 659, 699, 794
- \f@family 39, 660, 699, 707
- \f@series 655, 743
- \f@shape 655, 743
- \f@size 655, 661, 738, 746, 749, 750, 1624, 1660, 2452
- \font@name 39, 40, 664, 671, 672, 675, 678, 696, 697, 714, 774, 775, 776, 777, 778
- \get@external@font 690
- \getanddefine@fonts 772, 786
- \Hy@driver 1553
- \Hy@unicodetrue 58
- \HyPsd@ConvertToUnicode 58
- \HyPsd@LoadUnicode 59
- \HyPsd@pdfencoding 58, 58
- \HyPsd@ToBigChars 58
- \if@mainmatter 2272, 2307, 2498, 2694, 2695
- \if@noskipsec 2190
- \if@twocolumn 2288
- \if@twoside 2475, 2482
- \ifin@ 788, 796
- \itshape 37
- \ltj@@al@getanddefine@fonts 772, 789
- \ltj@@getjfontnumber 759
- \ltj@@IsFontJapanese 767, 771, 787
- \ltj@@ja@getanddefine@fonts 773, 788
- \ltj@@mathJapaneseFonts 771, 795
- \ltj@@set@stackfont 781, 782, 783
- \ltj@allalchar 603, 621
- \ltj@curjfont 40, 61
- \ltj@pickup@altfont@aux 42, 744
- \ltj@pickup@altfont@copy 42, 757
- \ltj@setpar@global 780
- \ltj@temp 788, 789, 790
- \ltj@tempcnta 787, 795
- \ltj@tempcntc 42, 763
- \m@ne 2204, 2214, 2271, 2306, 2861, 2862, 2863, 2864, 2865, 2866
- \math@bgroup 793
- \math@egroup 797
- \math@fonts 780
- \nfss@catcodes 43
- \p@ 1624, 1660, 1778, 1779, 1780, 1786, 1787, 1788, 1790, 1791, 1797, 1798, 1799, 1801, 1802, 1816, 1817, 1818, 1824, 1825, 1826, 1828, 1829, 1835, 1836, 1837, 1839, 1840, 2241, 2286, 2287, 2296, 2308, 2315, 2320, 2333, 2452, 2551, 2563, 2565, 2566, 2654
- \pickup@font 39
- \pickup@jfont 687, 775, 777, 779
- \protected@edef 2036, 2038, 2346, 2505
- \ps@fancy 77, 2489, 2491, 2493, 2497, 2500
- \ps@headings 77, 2474, 2476, 2480, 2483
- \reset@font 441, 1098
- \rmfamily 37
- \scriptfont@name 777
- \sf@size 776
- \sffamily 37
- \size@update 65, 66, 1903, 1917
- \split@name 749
- \ssf@size 778

- `\subst@correction` 710, 717
- `\textfont@name` 775
- `\tf@size` 774
- `\ttfamily` 37
- `\use@mathgroup` 792
- `\verbatim@font` 1455, 1456
- `\wrong@fontshape` 39
- `\x@protect` 76
- `\XeTeXglyphbounds` 96
- `\z@` 1779, 1787, 1798, 1817, 1825, 1836,
2194, 2212, 2251, 2298, 2304, 2322, 2328, 2349, 2384, 2630
- tex commands:
 - `\tex_divide:D` 66, 1951, 1992
 - `\tex_endlinechar:D` 452, 454
 - `\tex_font:D` 3575
 - `\tex_iftrue:D` 2695
 - `\tex_ignorespaces:D` .. 913, 1051, 1056, 1708, 1719, 1976
 - `\tex_lowercase:D` 52, 137, 613, 1160
 - `\tex_uppercase:D` 1528
- textfont@name commands:
 - `\textfont@name:` 781
- `\the` 763, 3250, 3253, 3254, 3273, 3277
- `\thechapter` ... 14, 19, 19, 2274, 2277, 2308, 2481, 2497, 2560
- `\theparagraph` 2596
- `\thepart` 2206, 2215, 2231, 2241, 2539
- `\thesection` 14, 19, 19, 2474, 2483, 2492, 2500, 2572
- `\thesubparagraph` 2604
- `\thesubsection` 2476, 2494, 2580
- `\thesubsubsection` 2588
- `\thispagestyle` 72, 2228, 2295, 2319
- three commands:
 - `\c_three` 299, 1996, 2619, 3592
- `\tiny` 1807, 1845
- tl commands:
 - `\tl_clear:c` 2169
 - `\tl_clear:N` 1385, 1389, 1694, 1859, 2172, 2179
 - `\tl_const:cx` 2106, 2112
 - `\tl_const:Nn`
.. 430, 807, 818, 1087, 1088, 1089, 1090, 1530, 1539, 2686
 - `\tl_const:Nx` 1136, 1913
 - `\tl_gset:Nn` 1110, 1526, 1536, 1537
 - `\tl_gset:Nx` 468, 471, 802, 1470, 1499, 2820
 - `\tl_gset_eq:NN` 895, 1494
 - `\tl_gset_rescan:Nnn` 1117
 - `\tl_head:N` 1863
 - `\tl_if_blank:nTF` 1202, 1298, 1299, 1403, 1404, 1418, 1436
 - `\tl_if_blank_p:n` 1171
 - `\tl_if_empty:cT` 2519
 - `\tl_if_empty:NF` 650
 - `\tl_if_empty:NTF` 2116, 2785
 - `\tl_if_eq:NNT` 1466
 - `\tl_if_eq:NNTF` 1906
 - `\tl_if_exist:NF` 429, 431, 432, 433, 1086, 1088, 1089, 1090
 - `\tl_if_exist:NTF` 2082, 2085, 2688
 - `\tl_map_function:nN` 51
 - `\tl_map_inline:Nn` 1314
 - `\tl_map_inline:nn` 1860
 - `\tl_new:c` 2104, 2105
 - `\tl_new:N` 24, 142, 431, 432, 433, 652, 653, 741, 801, 855,
881, 901, 902, 923, 1250, 1391, 1487, 1914, 1930, 2419, 2420
 - `\tl_put_left:Nn` 1674
 - `\tl_put_right:Nn` ... 1431, 2145, 2147, 2149, 2151, 2153
 - `\tl_put_right:Nx` 1398, 1863, 1864, 2130
 - `\tl_replace_all:Nnn` 1168, 1206
 - `\tl_set:cn` 2167, 2170
 - `\tl_set:Nn` 204, 205, 404, 421, 422, 423, 640,
642, 654, 742, 839, 906, 918, 1167, 1203, 1205, 1246,
1419, 1423, 1693, 1924, 1925, 1931, 2736, 2988, 2989, 2990
 - `\tl_set:Nx` 143, 400, 464, 693, 1042, 1489, 1979
 - `\tl_set_eq:Nc` 357
 - `\tl_set_eq:NN` 659, 660, 707, 812, 907, 932, 1353, 1365, 1915
 - `\tl_trim_spaces:n` 1382
 - `\today` 12, 12, 2039, 2044, 2047, 2052
 - today 12, 68
 - token commands:
 - `\token_get_replacement_spec:N` 62, 82
 - `\token_to_str:N` 123, 960,
961, 962, 964, 971, 972, 973, 1035, 1727, 2079, 2080, 3582
 - `\toks` 3250, 3251, 3253, 3254, 3273, 3277
 - `\topsep` 1790, 1801, 1828, 1839
 - `\TrimSpaces` 376
 - `\ttdefault` 1476
 - `\ttfamily` 438, 439, 1095, 1096
 - two commands:
 - `\c_two` 145, 1997, 2452, 2611, 2627
 - `\c_two_hundred_fifty_six` 467, 470
 - `\typeout` 2274, 2275

U

 - um commands:
 - `\um_cs_set_eq_active_char:Nw` 630, 631
 - `\undefined` 3275
 - `\updefault` ... 809, 814, 816, 820, 822, 1138, 1140, 1143, 1147
 - `\uppercase` 35, 35
 - use commands:
 - `\use:c` 661, 699, 700, 750, 823, 873, 1366
 - `\use:n` 1107, 1111, 1488, 1601, 1669, 2400, 2457
 - `\use:x`
. 60, 447, 637, 749, 854, 990, 1208, 1382, 1737, 2178, 2794
 - `\use_i:nn` 38
 - `\use_i:nnn` 245
 - `\use_ii:nn` 40
 - `\use_ii:nnn` 147, 250
 - `\use_iii:nnn` 255
 - `\use_none:n` 890, 1110, 1243, 1602, 2034
 - UTF8 6, 29

V

 - `\verse` 82, 2750, 2751
 - `\vskip` 2222, 2223,
2241, 2256, 2257, 2308, 2315, 2316, 2333, 2334, 2551, 2563
 - `\vspace` 2296, 2303, 2320, 2327

W			
winfonts	22	\youyuan	9, 9, 3241
\write	3284	Z	
X		zero commands:	
\x	3252, 3255, 3256, 3257, 3258, 3259, 3260, 3261, 3262, 3263, 3264, 3265, 3266, 3267, 3268, 3269, 3270, 3271, 3278, 3282, 3286	\c_zero	155, 169, 608, 612, 1952
\xdef	774, 776, 778, 3285	\c_zero_dim	1944, 1956, 1971, 1987, 1994, 2156
\xeCJKsetup	546, 1639, 1703, 1706, 1709	\c_zero_fp	1913
xetex commands:		\c_zero_skip	2154, 2155
\xetex_if_engine:F	3548	\zhdigits	11, 11, 2038
\xetex_if_engine:TF	405, 3188	zhmap	6, 22, <u>30</u>
\XeTeXcharglyph	3579	\zhnum	2032
\XeTeXglyphbounds	3599	\zhnumber	11, 11, 11, 12, 12, 2036
Y		\zhnumsetup	2048, 2053
\yahei	9, 9, 3243	\zhtoday	2047, 2052
		\zihao	5, 9, 9, 9, <u>62</u> , 1718, 1727
		\ziju	9, 9, <u>67</u> , 1975
		\zw	55, 55, 60, 591, 1447, 1452, 1628, 1661