

CT_EX 套件手册

ctex.org

2015/05/16 v2.0.2*

简介

CT_EX 套件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 套件支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX 和 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 和中文文档类 ctexart、ctexrep、ctexbook 实现。

*ctex-kit rev. 3216ced.

目录

第 1 节 介绍	3	第 11 节 CT_EX 套件的配置文件	24
第 2 节 简明教程	4	11.1 修改宏包默认选项	25
2.1 CT _E X 套件的组成	4	11.2 宏包载入后的配置	25
2.2 使用 CT _E X 文档类	4	11.3 配置标题中文翻译	26
2.3 使用 ctex 宏包	4	11.4 自定义字体集	26
第 3 节 依赖与安装	5	第 12 节 对旧版本的兼容性	26
第 4 节 宏包选项与 \ctexset 命令	6	12.1 CT _E X 0.8a 及以前的版本	26
第 5 节 编译方式、编码与中文字库	6	12.2 CT _E X 0.9–CT _E X 1.0d	26
5.1 编译方式	6	12.3 CT _E X 1.02c 以后的 SVN 开发版	28
5.2 中文编码	7	第 13 节 开发人员	28
5.3 中文字库	7	第 14 节 参考文献	28
第 6 节 排版格式设定	9	第 15 节 代码实现	29
6.1 文档默认字号	9	15.1 内部函数与变量	29
6.2 章节标题风格	10	15.2 宏包选项	35
6.3 排版方案选项	10	15.3 用户设置接口	40
第 7 节 文档汉化	12	15.4 特定引擎支持与设置	40
7.1 日期汉化	12	15.5 字距与缩进	71
7.2 文档标题汉化	12	15.6 中文数字与日期	74
7.3 页面格式设置与汉化	13	15.7 其它中文标题定义	75
第 8 节 章节标题格式设置	15	15.8 中文化的标题结构	76
第 9 节 实用命令	21	15.9 chinese 方案的其他设置	89
9.1 字号与间距	21	15.10 中文字号	89
9.2 中文数字转换	22	15.11 更新行距	93
9.3 杂项	23	15.12 其它功能	93
第 10 节 Lua_AT_EX 下的中文支持方式	23	15.13 载入中文字体	94
10.1 Lua _A T _E X 下替代字体的设置	23	15.14 宏包配置文件	96
		15.15 字体定义文件	96
		第 16 节 版本历史	112
		第 17 节 代码索引	112

第 1 节 介绍

历史

CT_EX 套件的源头有两个：一是王磊编写的 `cjkbook` 文档类，二是吴凌云编写的 `GB.cap`。这些工作没有经过认真、系统的设计，也没有用户文档，不利于维护和改进。

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

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

2014 年 3 月，为了适应 L^ATeX 的最新发展，特别是 L^ATeX3 的逐渐成熟，李清用 L^ATeX3 重构了整个宏包的代码，并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理，升级版本号为 2.0，并改称 CT_EX 套件。

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 GitHub²。

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国语言支持，特别是对多字节的中日韩语言的支持。这使得 T_EX 以至后来的 L^ATeX 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后，中文用户使用 L^ATeX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文语言习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 CT_EX 套件的目的一就是解决这些 L^ATeX 文档的汉化难题。

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

CT_EX 套件的许多实现细节离不开热心朋友们在 bbs.ctex.org 论坛上的讨论，在此对参与讨论的朋友们表示感谢。

关于套件名字的说明

CT_EX 之名是英文单词 China（中国）或 Chinese（中文）的首字母“C”与“T_EX”结合而成的。在纯文本环境下，该名字应写作“CT_EX”。

CT_EX 套件是由 CT_EX 社区发起并维护的 LaTeX 宏包和文档类的集合。社区另有发布名为 CT_EX 套装的 T_EX 发行版，与本文档所述的 CT_EX 套件并非是一事物。

`ctex` 则是本套件中的 `ctex.sty` 的名字。这一完全小写的名称，在过去也被用来指代整个 CT_EX 套件，不过现在则特指 `ctex.sty` 这一宏包。

¹<http://code.google.com/p/ctex-kit/>

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

³比如：pdf_TE_X 引擎下的 CJK、zhmCJK 宏包、X_YTeX 引擎下的 xeCJK 宏包和 Lua_TE_X 引擎下的 LuaTeX-ja 宏包

第 2 节 简明教程

2.1 C_TE_X 套件的组成

为了适应用户不同的需求,我们将 C_TE_X 套件的主要功能分散在三个中文文档类和三个宏包当中,具体的组成见表 1。

表 1 C_TE_X 套件的组成

类别	文件	说明
文档类	ctexart.cls	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
	ctexrep.cls	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
	ctexbook.cls	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
宏包	ctex.sty	提供全部功能,但默认不开启章节标题设置功能,需要使用 <code>heading</code> 选项来开启
	ctexcap.sty	过时宏包,仅作兼容使用,相当于开启章节标题设置功能的 <code>ctex.sty</code>
	ctexsize.sty	定义和调整中文字号,可在 <code>ctex</code> 宏包或 C _T E _X 中文文档类之外单独调用

2.2 使用 C_TE_X 文档类

如果用户需要在标准文档类的基础上添加中文支持和中文版式支持,我们建议用户使用 C_TE_X 套件提供的三个中文文档类。

C_TE_X 套件提供了三个中文文档类: `ctexart`、`ctexrep` 和 `ctexbook`,分别对应 L^AT_EX 的标准文档类 `article`、`report` 和 `book`。使用它们的时候,需要将涉及到的所有源文件使用 UTF-8 编码保存⁴。

例 1

```
\documentclass[UTF8]{ctexart}
\begin{document}
中文文档类测试。你需要将所有源文件保存为 UTF-8 编码。

你可以使用 XeLaTeX 或 LuaLaTeX 编译,也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。
\end{document}
```

2.3 使用 ctex 宏包

用户在使用非标准文档类时,如果需要添加中文支持或中文版式支持,则可以使用 `ctex` 宏包。

以下是使用 `beamer` 文档类编写中文演示文稿的一个示例。

例 2

```
\documentclass{beamer}
\usepackage[UTF8]{ctex}
\begin{document}
\begin{frame}{中文演示文档}
\begin{itemize}
\item 你需要将所有源文件保存为 UTF-8 编码
\item 你可以使用 XeLaTeX 或 LuaLaTeX 编译
\end{itemize}
\end{frame}
\end{document}
```

⁴使用 (pdf)L^AT_EX 时也能够使用 GBK 编码,但不推荐。(见 5.2 节)

```

\item 也可以使用 (pdf)LaTeX 编译
\item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\end{itemize}
\end{frame}
\end{document}

```

有些文档类是建立在 L^AT_EX 标准文档类之上开发的。这时,给 ctex 宏包加上 heading 选项,可以将章节标题设置为中文风格。

例 3

```

\documentclass{ltxdoc}
\usepackage[UTF8, heading = true]{ctex}
\begin{document}
\section{简介}
章节标题中文化的 \LaTeX{} 手册。
\end{document}

```

第 3 节 依赖与安装

CT_EX 是一个 L^AT_EX 宏包,只有一个源文件 ctex.dtx。使用不同的编译方式时,CT_EX 依赖的宏包略有不同。在安装 CT_EX 套件之前,请确保你的系统里已经正确安装了这些宏包。CT_EX 依赖宏包的详情叙述如下:

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 套件,它们还依赖
 - etex 宏包。
- ifpdf 宏包,属于 oberdiek 套件。
- etoolbox 宏包。
- everyisel 宏包,属于 ms 套件。
- zhnumber 宏包。
- ⇒ 以上是各种编译方式都必需的依赖项。
- CJK 套件,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 pdfL^AT_EX 或 L^AT_EX + DVIPDFM_x 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。
- ⇒ 以上是使用 X_YL^AT_EX 编译时的依赖项。
- LuaTeX-ja 套件,它还依赖
 - luaotfload 宏包,它还依赖 luatexbase 宏包。
 - xkeyval 宏包。
- ⇒ 以上是使用 LuaL^AT_EX 编译时的依赖项。

CT_EX 依赖的各个宏包和套件均已被 T_EX Live 和 MiK_TE_X 收录。如果你是这两个 T_EX 发行版的用户,可以使用他们提供的宏包管理器来安装这些宏包,或将他们更新到最新版本。如果你不使用这两个发行版,请参考你使用的 T_EX 发行版和相关宏包的说明文档。

出于一些原因,zhmCJK 尚未被收入 T_EX Live 和 MiK_TE_X。因此,若你希望使用 zhmCJK 作为 CT_EX 套件的底层中文支持方式,需要自行安装该宏包。zhmCJK 的安装较为复杂。我们

建议你从 CTAN 下载 zhmcjk 的 [TDS 安装包](#), 按目录结构将文件复制到 TeX 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, 可以参照其 [宏包手册](#) 中第 3 节的指导。

CTeX 的正式版本也已被 TeX Live 和 MiKTeX 收录。因此, 使用这两个发行版的用户可以如常使用宏包管理器来安装。如果你不使用者两个发行版, 我们建议你从 CTAN 下载 CTeX 的 [TDS 安装包](#) 按目录结构将文件复制到 TeX 发行版的本地 TDS 根目录, 然后刷新文件名数据库。

第 4 节 宏包选项与 \ctexset 命令

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

不过, CTeX 套件也提供了一系列选项。用户可以使用这些选项来控制 CTeX 套件的行为。具体来说, 这些选项里, 有的以传统的方式提供, 也有的以 $\langle key \rangle = \langle value \rangle$ 的形式提供。对于以键值对形式提供的选项, 在下面的说明中使用**粗体**来表示 CTeX 的默认设置。

另一方面, 在这些选项里, 有些选项只能作为宏包/文档类选项, 需要在引入宏包/文档类的时候指定; 有些选项只能通过 CTeX 套件提供的用户接口 `\ctexset` 来设定; 也有一些选项既可以作为宏包/文档类选项, 也可以通过 `\ctexset` 来设定。后续文档中将在使用说明中对每个选项的使用方式做说明。

`\ctexset` `\ctexset {<键值列表>}`

是 CTeX 套件的通用控制命令, 用来在宏包载入后控制宏包的各项功能。`\ctexset` 的参数是一个键值列表, 以通用的接口完成各项设置。

`\ctexset` 的参数是一组由逗号分隔的选项列表, 列表中的选项通常是一个 $\langle key \rangle = \langle value \rangle$ 格式的定义。例如设置摘要与参考文献标题名称(7.2 节)就可以使用:

例 4

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

`\ctexset` 采用 L^AT_EX 3 风格的键值设置, 支持不同类型的选项与层次化的选项设置, 相关示例见 8 节。

第 5 节 编译方式、编码与中文字库

5.1 编译方式

CTeX 套件会根据用户使用的编译方式⁵, 在底层选择不同的中文支持方式(见表 2)。

不同的编译方式和中文支持方式会在一定程度上影响 CTeX 套件的行为, 比如宏包对编码的处理。在用户使用 X_qL^AT_EX 及 LuaL^AT_EX 编译时, CTeX 套件使用(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, 宏包默认使用 GBK 编码。用户需要保证编译方式、源文件编码、宏包编码选项三者一致。关于宏包编码选项, 可以参考 5.2 节。

⁵L^AT_EX、pdfL^AT_EX、X_qL^AT_EX 以及 LuaL^AT_EX。

表 2 CT_EX 套件的中文支持方式

(pdf)L _A T _E X	X _Y L _A T _E X	LuaL _A T _E X
CJK	xeCJK	LuaTeX-ja

除去文档编码之外,选择不同的编译方式还可能影响 CT_EX 套件对字体选择、空格处理、标点处理的处理。具体的影响将在本文档后续内容中进行阐述。

5.2 中文编码

GBK
UTF8

该选项只能在调用 CT_EX 宏包/文档类时作为选项使用

指明编写文档时使用的编码格式。CT_EX 套件无法检测用户编写文档时使用的编码格式,因此需要用户自行指定编码。我们建议用户总是使用 UTF-8 编码,并显式指定 UTF8 选项,并使用 X_YL_AT_EX 或 LuaL_AT_EX 编译。

使用 X_YL_AT_EX 或 LuaL_AT_EX 编译时,CT_EX 套件强制使用 UTF-8 编码,此时 GBK 选项无效。使用 (pdf)L_AT_EX 编译时,CT_EX 套件默认启用 GBK 选项;不过,用户也可以显式声明 UTF8 选项,使 CT_EX 套件工作在 UTF-8 编码下。

5.3 中文字库

以往,为 L_AT_EX 文档配置中文支持是一件相当繁琐的事情。默认情况下,CT_EX 套件能自动检测用户使用的编译方式(参见 5.1 节)和操作系统⁶,选择合适的底层支持和字库,从而简化配置过程。自动配置的情况参见表 3。

表 3 CT_EX 套件自动配置字体策略

	Mac OS X	Windows New ¹	Windows Old ²	其他
X _Y L _A T _E X	xeCJK 华文字库	xeCJK 中易字库 + 微软雅黑	xeCJK 中易字库	xeCJK Fandol 字库 ³
LuaL _A T _E X ⁴	LuaTeX-ja 华文字库	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja 中易字库	LuaTeX-ja Fandol 字库
pdfL _A T _E X	不可用	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	不可用
L _A T _E X + DVIPDFM _x	不可用	CJK + zhmetrics 中易字库 + 微软雅黑	CJK + zhmetrics 中易字库	CJK + zhmetrics Fandol 字库

¹ Windows Vista 及以后的 Windows 操作系统。

² Windows XP 及以前的 Windows 操作系统。

³ 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>。

⁴ LuaL_AT_EX 编译时使用 LuaTeX-ja 宏包。对此,第 10 节有特别说明。

通常,由 CT_EX 套件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。

⁶CT_EX 套件现在能够识别 Mac OS X 系统以及 Windows 系统。

zhmap `zhmap = <true|false|zhmCJK>`
 该选项只能在调用 CTeX 宏包/文档类时作为选项使用,并且只在使用 $\text{pdf}\mathcal{L}\text{A}\text{T}\text{E}\text{X}/\mathcal{L}\text{A}\text{T}\text{E}\text{X}$ 编译时有意义
 指定字体映射机制。

选项为 `true` 时,使用 `zhmetrics` 宏包,将 CJK 字库通过 `\special` 命令映射到 `.ttf` 文件;
 选项为 `false` 时,使用传统的 CJK 字库(Type 1)⁷;选项为 `zhmCJK` 时,将载入 `zhmCJK` 宏包⁸,
 由 `zhmCJK` 宏包提供从 CJK 字库到 `.ttf` 的映射。

fontset `fontset = <none|adobe|fandol|founder|mac|ubuntu|windows|windowsnew|windowsold...>`
 该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置
 指定 CTeX 套件加载的字库。

如果没有指定 `fontset` 的值, CTeX 套件将自动检测用户使用的操作系统,配置相应的字体(参见表 3)。

CTeX 预定义了以下六种中文字库。

<code>adobe</code>	使用 Adobe 公司的四款中文字体,不支持 $\text{pdf}\mathcal{L}\text{A}\text{T}\text{E}\text{X}$ 。
<code>fandol</code>	使用 Fandol 中文字体,不支持 $\text{pdf}\mathcal{L}\text{A}\text{T}\text{E}\text{X}$ 。
<code>founder</code>	使用方正公司的中文字体。
<code>mac</code>	使用 Mac OS X 系统下的华文字体,不支持 $(\text{pdf})\mathcal{L}\text{A}\text{T}\text{E}\text{X}$ 。
<code>ubuntu</code>	使用 Ubuntu 系统下的文泉驿和文鼎字体。
<code>windows</code>	使用简体中文 Windows 系统下的中文字体,自动判断 Windows 系统版本,采用 <code>windowsnew</code> 或 <code>windowsold</code> 的设置。
<code>windowsnew</code>	使用简体中文 Windows Vista 或之后系统下的中易字体和微软雅黑字体。
<code>windowsold</code>	使用简体中文 Windows XP 或之前系统下的中易字体。

注意:使用 $(\text{pdf})\mathcal{L}\text{A}\text{T}\text{E}\text{X}$ 编译的时候,若设置 `zhmap = false`(比如需要使用 $\mathcal{L}\text{A}\text{T}\text{E}\text{X} + \text{DVIPS}$ 编译),则需要按照传统方式⁹在本地安装好 CJK 字体。

如果不想使用 CTeX 预定义的中文字库,可以设置 `fontset` 为下述值之一。

<code>none</code>	不配置中文字体,需要用户自己配置。
<code><name></code>	这里 <code><name></code> 为自定义的名字。 CTeX 套件将载入名为 <code>ctex-fontset-<name>.def</code> 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 <code>ctex-fontset-<name>.def</code> 的文件,在这个文件里面自定义中文字体。然后通过使用 <code>fontset=<name></code> 选项来调用它。字体配置文件的具体写法可以参考 CTeX 套件 <code>fontset</code> 目录下的字体配置文件。

注意:如果希望使用 `\ctexset` 在导言区指定字库,则需要先在宏包/文档类选项中指定 `fontset = none`。例如:

例 5

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

CTeX 套件预定义的中文字库还定义了一些字体命令。

<code>\songti</code>	宋体,CJK 等价命令 <code>\CJKfamily{zh-song}</code> 。
----------------------	--

⁷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。

⁸`zhmCJK` 宏包基于 `zhmetrics` 和 `CJK` 宏包,提供与 `xeCJK` 宏包类似的用户接口。

⁹可以使用 `zhmetrics` 宏包提供的脚本 `CTeXFonts.lua`。

`\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}`。

其中 `\fangsong` 在 ubuntu 字库中没有定义。在 windows 和 founder 字库中,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 windowsnew 字库中,还有 `\yahei`。

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

第 6 节 排版格式设定

6.1 文档默认字号

`zihao = <-4|5|false>`
 该选项只能在调用 \TeX 宏包/文档类时作为选项使用

将文章默认字号(`\normalsize`)设置为小四号字或五号字,`\false` 禁用本功能。这两个选项可以用于三个 \TeX 文档类和 `ctex` 宏包,也可以用于 `ctexsize` 宏包。

`scheme = chinese` 时,默认值为 5,即设置 `\normalsize` 为五号字。

`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 4)。

表 4 标准字体命令与字号的对应

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

`10pt` 该选项只能在调用 \TeX 宏包/文档类时作为选项使用

`11pt`
`12pt`

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

标准文档类的其他选项在 \TeX 文档类中依旧有效。例如,设置纸张大小和方向的 `a4paper` 和 `landscape`,设置单双面的 `oneside` 和 `twoside` 等。 \TeX 会将这些选项传给

标准文档类¹⁰。

6.2 章节标题风格

heading `heading = <true|false>`
 该选项只能在调用 `ctex.sty` 时作为宏包选项使用

CT_EX 套件提供了一套用于修改文档章节标题格式的接口。该选项用于选择是否启用该功能。详细的设置方法请参见 8 节和 7.3 节。

CT_EX 套件提供的三个文档类总是启用该功能。如果在 `ctex.sty` 下启用该选项,将会检查当前是否使用 L^AT_EX 标准文档类。若然,则该选项将会使得 `ctex.sty` 宏包的行为和 CT_EX 套件提供的三个中文文档类完全一致;若不然,则会根据 `\chapter` 是否有定义来使用 `ctexbook` 或者 `ctexart` 的标题设置。

sub3section 该选项只能在调用 CT_EX 宏包/文档类时作为选项使用

sub4section

会修改 `\paragraph` 和 `\subparagraph` 的格式。

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

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

具体格式可参考 8 节中的 `afterskip` 等选项。

注意,上述两个选项只有在 `heading` 选项启用的时候才有意义。亦即,只有在使用三个 CT_EX 文档类或启用了 `heading` 的 `ctex.sty` 的时候才有意义。

6.3 排版方案选项

scheme `scheme = <chinese|plain>`
 该选项只能在调用 CT_EX 宏包/文档类时作为选项使用

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

chinese 调整默认字号为五号字,并调整行距为 1.3; 汉化文档中的标题名字(如“图”、“表”、“目录”和“参考文献”,见 7.2 节);在 `heading = true` 的情况下¹¹(6.2 节),还会将章节标题的风格修改为中文样式(见 8 节)。当 `ctex` 宏包与标准文档类及其衍生文档类联用时,会载入 `indentfirst` 宏包,以实现章节标题后的段首缩进。

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

punct `punct = <quanjiao|banjiao|kaiming|CCT|plain>`
 该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置

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¹²用占一个汉字宽度,标号和句内点号占半个汉字宽度;

¹⁰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。

¹²标点符号分为标号与点号。点号分为两类,一共七种:句末点号有句号、问号和叹号;句内点号有逗号、顿号、冒号和分号。

CCT CCT 式:所有标点符号的宽度略小于一个汉字宽度;
plain 原样(不调整标点间距)。

space `space = <true|false|auto>`
该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置
是否在生成的 PDF 中保留汉字后面的空格。

true 总是保留汉字后的空格。此时,用户需要自行在行尾加上 % 处理换行产生的空格¹³。
false 总是忽略掉汉字后面的空格,不论汉字后是什么(使用 (pdf)L^AT_EX 编译时);等同于 auto 的效果(使用 X_YL^AT_EX 编译时)。不建议使用该选项。
auto 根据空格后面的情况决定是否保留:如果空格后面是汉字,则忽略该空格,否则保留。

例如,使用

例 6

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

将得到“汉字 分词 技术 English”;使用

例 7

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

则会得到“汉字分词技术 English”。

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

linespread `linespread = <数值>`
该选项只能在调用 C_TE_X 宏包/文档类时作为选项使用
接受一个浮点数值,设置行距倍数。

`scheme = chinese` 时, 默认值为 1.3, 即 1.3 倍行距。此时, 相邻两行的基线 (`\baselineskip`) 距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。

`scheme = plain` 时, C_TE_X 套件默认不调整行距倍数, 文档中的行距由所选文档类和其他宏包或用户设置决定。

autoindent `autoindent = <true|false|数值|带单位的数值>`
该选项既可以作为宏包/文档类选项使用,也可以使用 `\ctexset` 来设置

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

`<数值 | 带单位的数值>` 用于设置段首缩进的长度。如果不带单位,则默认单位是单个汉字字宽 `\ccwd`;如果带单位,则使用该单位。

`autoindent = true` 等价于设置 `autoindent = 2`。

当 `autoindent = false` 时,可以设置固定长度的段首缩进。如设置每段缩进 40 点:

例 8

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

¹³L^AT_EX 将单个换行视作一个空格。

linestretch `linestretch = <数值或长度>`
 该选项只能用 `\ctexset` 来设置

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

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

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

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

第 7 节 文档汉化

7.1 日期汉化

CT_EX 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 5 月 16 日”。

today `today = <small|big|old>`
 该选项只能用 `\ctexset` 来设置

该选项用来控制 `\today` 命令的输出格式:

- small** 效果为“2015 年 5 月 16 日”。使用阿拉伯数字和汉字的日期格式。
big 效果为“二〇一五年五月十六日”。使用全汉字的日期格式。
old 效果为“May 16, 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}
```

CT_EX 宏包的中文日期功能实际上是调用 `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。

7.2 文档标题汉化

这里主要介绍由宏包 `scheme` 选项(6.3 节)控制的文档标题汉化功能。

设置文档标题名的示例可见例 4。下面的选项(如 `contentsname`)主要用来重新定义与选项同名的宏(如 `\contentsname`)的定义。

contentsname `contentsname = <名字>`
 该选项只能用 `\ctexset` 来设置

设置目录标题名 `\contentsname`。中文默认为“目录”。

listfigurename `listfigurename = <名字>`
 该选项只能用 `\ctexset` 来设置

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

<code>listtablename</code>	<code>listtablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置表格目录标题名 <code>\listtablename</code> 。中文默认为“表格”。
<code>figurename</code>	<code>figurename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置图片环境标题名 <code>\figurename</code> 。中文默认为“图”。
<code>tablename</code>	<code>tablename = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置表格环境标题名 <code>\tablename</code> 。中文默认为“表”。
<code>abstractname</code>	<code>abstractname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置摘要 <code>abstract</code> 环境标题名 <code>\abstractname</code> 。中文默认为“摘要”。注意 <code>book</code> 类没有摘要，该选项无效。
<code>indexname</code>	<code>indexname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置索引标题名 <code>\indexname</code> 。中文默认为“索引”。
<code>appendixname</code>	<code>appendixname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置附录标题名 <code>\appendixname</code> 。中文默认为“附录”。
<code>bibname</code>	<code>bibname = <名字></code> 该选项只能用 <code>\ctexset</code> 来设置 设置参考文献标题名 <code>\refname</code> (对 <code>article</code>) 或 <code>\bibname</code> (对 <code>report</code> 和 <code>book</code>)。中文默认为“参考文献”。

7.3 页面格式设置与汉化

当使用了 `CTEX` 的文档类或是用 `ctex` 宏包加载了 `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` 宏包同时也为这些自定义页面格式的包提供了以下宏供使用:

- `\CTEXthechapter`、`\CTEXthesection` 等章节编号(见 8 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 [1, Chapter 23]、[2, §4.3, §4.4] 等书籍。

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

例 10

```
\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”。

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

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

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

例 11

```
\documentclass{ctexart}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
}
\usepackage{fancyhdr}
\fancyhf{}
\lhead{\textnormal{\kaishu\rightmark}}
\rhead{--\ \thepage\ --}
\pagestyle{fancy}
% \sectionmark 的重定义需要在 \pagestyle 之后生效
\renewcommand\sectionmark[1]{%
  \markright{\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 8 节 章节标题格式设置

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(6.2 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)}

该选项只能用 \ctexset 来设置

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 12

```
\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。

表 5 name 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 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	{}	{}	

```

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

```

number = {<数字输出命令>}
 该选项只能用 `\ctexset` 来设置

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 13

```

\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 \LaTeX 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。)

表 6 number 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 的默认值	原 <code>\the<标题></code> 等价定义
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>\thesubsubsection</code>	<code>\thesubsection.\arabic{subsubsection}</code>
paragraph	同右	<code>\theparagraph</code>	<code>\thesubsubsection.\arabic{paragraph}</code>
subparagraph	同右	<code>\thesubparagraph</code>	<code>\theparagraph.\arabic{subparagraph}</code>

```

\CTEXthepart
\CTEXthechapter
\CTEXthesection
\CTEXthesubsection
\CTEXthesubsubsection
\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 `scheme = chinese` 时, 默认章编号输出格式就是 `\CTEXthechapter`, 形如“第一章”。

这组宏在 \CTEX 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 14

```

\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}}

```

该选项只能用 `\ctexset` 来设置

`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 15

```

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

```

表 7 `format` 选项的默认设置

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

```

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}}

```

该选项只能用 `\ctexset` 来设置

`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。

表 8 `nameformat` 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 的默认值
part (article)	同右	<code>\Large\bfseries</code>
part	同右	<code>\huge\bfseries</code>
chapter	同右	<code>\huge\bfseries</code>
section	同右	<code>{}</code>
subsection	同右	<code>{}</code>
subsubsection	同右	<code>{}</code>
paragraph	同右	<code>{}</code>
subparagraph	同右	<code>{}</code>

```
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)}
该选项只能用 \ctexset 来设置
```

`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 16

```
\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}
该选项只能用 \ctexset 来设置
```

`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。

表 9 `aftername` 选项的默认设置

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

```
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)}
该选项只能用 \ctexset 来设置
```

`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。

需要注意的是, `titleformat` 选项的最后一个格式命令可以是一个有一个参数的命令。这个命令的参数接受标题内容,用于实现特殊效果。例如,实现多行标题的居中悬挂对齐:

例 17

```
\usepackage{varwidth} %% 提供 varwidth 环境
\ctexset{
  chapter/name = {第,回},
  chapter/titleformat = \chaptertitleformat
```

```

}
\newcommand\chaptertitleformat[1]{%% 以标题内容为参数
  \begin{varwidth}[t]{.7\linewidth}#1\end{varwidth}}
.....
\chapter{情中情因情感妹妹\错里错以错劝哥哥}

```

上面的代码可以做出类似这样的章标题效果：

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

表 10 titleformat 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 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	{}

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

beforeskip = {<弹性间距>}
该选项只能用 \ctexset 来设置

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 6.2 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。

表 11 beforeskip 选项的默认设置

标题名	scheme = chinese 的默认值	scheme = plain 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

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

```
afterskip = {<弹性间距>}
该选项只能用 \ctexset 来设置
```

`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 6.2 节)后,则这两级标题会改为排在不同段。

`afterskip` 选项的默认值,在 `scheme` 选项的不同取值下相同。

表 12 afterskip 选项的默认设置

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 indent 选项的默认设置

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

```
part/indent
chapter/indent
section/indent
subsection/indent
subsubsection/indent
paragraph/indent
subparagraph/indent
```

```
indent = {<缩进间距>}
该选项只能用 \ctexset 来设置
```

`indent` 选项用于设置章节标题本身的首行缩进。

`indent` 选项的默认值,在 `scheme` 选项的不同取值下相同。

例 18

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

```
part/pagestyle
chapter/pagestyle
```

```
pagestyle = {<页面格式>}
该选项只能用 \ctexset 来设置
```

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`)。

表 14 pagestyle 选项的默认设置

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

`appendix/name` name = {<前名字>,<后名字>}
name = {<前名字>}
该选项只能用 `\ctexset` 来设置

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 (7.2 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。

表 15 appendix/name 选项的默认设置

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

`appendix/number` number = {<数字输出命令>}
该选项只能用 `\ctexset` 来设置

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。

表 16 appendix/number 选项的默认设置

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

第 9 节 实用命令

9.1 字号与间距

`\zihao` `\zihao{<字号>}`

用于调整字号大小。其中 <字号> 的有效值共有 16 个, 如表 17 所示。使用 `\zihao` 命令调整字体大小时, 西文字号大小会始终和中文字号保持一致。

表 17 中文字号

〈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

\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。

9.2 中文数字转换

CT_EX 套件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。

\backslash zhdigits \backslash zhdigits $\{<number>\}$

将阿拉伯数字转换为中文数字串。

\backslash CTEXnumber \backslash CTEXnumber \backslash $\langle macro \rangle$ $\{<number>\}$

\backslash $\langle macro \rangle$ 必须是一个 \TeX 宏, 不需预先定义。 \backslash CTEXnumber 通过 \backslash zhnumber 将 $\langle number \rangle$ 转为中文数字, 最后将结果存储在 \backslash $\langle macro \rangle$ 里。对 \backslash $\langle macro \rangle$ 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 \backslash CTEXnumber, 直接使用 \backslash 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 \backslash CTEXnumber 将结果保存起来备用, 而不是每次使用时都用 \backslash zhnumber 现场转换一次。

\backslash CTEXdigits \backslash CTEXdigits \backslash $\langle macro \rangle$ $\{<number>\}$

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

9.3 杂项

\backslash CTeX 用于显示 \CTeX 标志。

第 10 节 Lua \LaTeX 下的中文支持方式

在 Lua \LaTeX 下, \CTeX 套件依赖 Lua \TeX -ja 宏包来完成中文支持。该宏包是日本 \TeX 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 \TeX 引擎下实现日本 \pTeX 引擎的(大部分)功能。它为了兼容 \pTeX 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, 因而 \CTeX 禁用了它在 \LaTeX 格式下的大部分设置, 只保留了必要的部分。同时修改了它的字体设置方式, 使得相关命令与 \xeCJK 宏包大致相同。

20150420 版以后的 Lua \TeX -ja 宏包开始支持竖排, 但 \CTeX 暂不支持竖排。

10.1 Lua \LaTeX 下替代字体的设置

AlternateFont \backslash setCJKfamilyfont $\{<family>\}$
 $[$
 AlternateFont =
 $\{$
 $\{<character\ range_1>\} [<alternate\ font\ features_1>] \{<alternate\ font\ name_1>\} ||$
 $\{<character\ range_2>\} [<alternate\ font\ features_2>] \{<alternate\ font\ name_2>\} ||$
 $\dots\dots$
 $\} ,$
 $\langle base\ font\ features \rangle$
 $] \{<base\ font\ name>\}$

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

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

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

一个 `\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 19

```
\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。

第 11 节 C_TE_X 套件的配置文件

C_TE_X 套件提供了不同的配置文件, 可以通过修改配置文件来改变 C_TE_X 套件的默认行为。

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

但在一些情况下, 直接修改配置文件仍是必要的, 例如:

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

与 C_TE_X 套件的源代码一样, 配置文件采用 L^AT_EX3 的语法编写。

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

例如对于 TeX Live,系统自带的配置文件就在 TeX Live 安装目录下的 texmf-dist/tex/latex/ctex/config/ 子目录下,可以修改它的副本,保存在本地 TDS 树的 texmf-local/tex/latex/ctex/ 目录下,或者用户 TDS 树的 ~/.texlive2015/texmf-var/tex/latex/ctex/ 目录下,作为本地/用户专有的配置文件。复制配置文件后需要运行 texhash 命令使本地配置文件生效。

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

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

11.1 修改宏包默认选项

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

例 20

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

如上例所示,宏包选项通常使用 L^AT_EX3 的 \keys_set:nn 命令完成键值设置,第一个参数是固定的子模块 ctex/option,第二个参数中是用户定义的新的默认宏包选项。

ctexopts.cfg 中的设置将在 CTeX 套件的开始处,定义过宏包选项之后,\ProcessKeysOptions 命令之前生效。最好只使用此配置文件修改宏包默认选项。

11.2 宏包载入后的配置

配置文件 ctex.cfg 将在宏包的末尾被载入生效。可以用它完成任意的设置,或是覆盖已有的定义。随系统安装的配置文件除版本信息外没有实际内容,注意配置文件中也使用 L^AT_EX3 语法。

例 21

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

例 22

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

11.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

11.4 自定义字体集

5.3 节介绍的用于 `fontset` 选项的自定义字库文件, 类似于 C_TE_X 套件的配置文件, 也应该与其他本地配置文件一起保存在本地 TDS 目录树下, 并可以配合 `ctexopts.cfg` 等配置文件使用。

第 12 节 对旧版本的兼容性

12.1 C_TE_X 0.8a 及以前的版本

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

第 2 版未考虑对这些很早版本的兼容性。

12.2 C_TE_X 0.9–C_TE_X 1.0d

在 2009 年在 `ctex-kit` 项目成立后, 新增了 X_YL_AT_EX 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 C_TE_X 宏包相对 1.02d 版本 (2014/06/09) 的兼容性。

第 2 版的 C_TE_X 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 C_TE_X 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。

C_TE_X 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 C_TE_X 宏包第 2 版中则完全不再支持 CCT。

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

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

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

`CCT` 相关选项已删除。
`CCTfont`

`indent` `indent` 什么也不做, `ctex` 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开;
`noindent` C_TE_X 文档类的相关功能由章节 `beforekip` 选项的值来确定。过时选项。

`noindent` 什么也不做, 也不会使章节首行停止缩进。需要使用 `\ctexset` 命令设置章节的 `beforekip` 选项为正数才能完成原来 `noindent` 选项的工作。另外, 设置 `scheme = plain` 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。

zhmap
nozhmap

zhmap 宏包选项增加了参数,扩充了功能,除了支持真假值参数外,还支持选择 zhmCJK 作为底层中文处理宏包。(5.3 节)

nozhmap 选项相当于 zhmap=false。过时选项。

winfonts
adobefonts
nofonts

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

另外,第 2 版 C_TE_X 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(5.3 节)。

punct
nopunct

旧版本中宏包 punct 选项没有参数,现在可以用参数设定标点风格(6.3 节)。原有无参形式的 punct 选项相当于 punct=quanjiao。

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

cap
nocap

原有的 cap 和 nocap 选项由新的 scheme 选项代替。(6.3 节)

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

space
nospace

新版本宏包 space 选项增加真假值参数。(6.3 节)

nospace 选项相当于 space=false,成为过时选项。

fancyhdr

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

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

hyperref

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

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

fntef

旧版本的 fntef 选项用于统一 CCTfntef 与 CJKfntef 的界面,新版本 C_TE_X 套件不再支持 CCT,而是直接载入 CJKfntef 或 xeCJKfntef 宏包并禁用其彩色设置。该选项是过时选项。

\CTEXunderdot
\CTEXunderline
\CTEXunderdblline
\CTEXunderwave
\CTEXsout
\CTEXxout
\CTEXfilltwosides

在调用 fntef 宏包选项的同时,旧版本 C_TE_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 C_TE_X 套件中已失去意义。

此外,在 pdf_TE_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。

\CTEXindent

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

\CTEXnoindent

设置 \parindent=0pt。过时命令。

\CTEXsetup

\CTEXsetup[<选项>]{<标题>}

相当于设置了 \ctexset{ <标题> = {<选项> } }。过时命令。

`\CTEXoptions` `\CTEXoptions[<选项>]`

相当于设置了 `\ctexset{<选项>}`。过时命令。

`\Chinese` `\Chinese[<counter>]`

新版套件中 `\chinese` 统一了旧版本中 `\chinese` 和 `\Chinese` 的功能。因此,该命令已过时。

`captiondelimiter`

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

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

例 23

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

12.3 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。

第 13 节 开发人员

- 吴凌云 (aloft@ctex.org)
- 江疆 (gzjjgod@gmail.com)
- 王越 (yuleopen@gmail.com)
- 刘海洋 (LeoLiu.PKU@gmail.com)
- 李延瑞 (LiYanrui.m2@gmail.com)
- 陈之初 (zhichu.chen@gmail.com)
- 李清 (sobenlee@gmail.com)
- 黄晨成 (liamhuang0205@gmail.com)

第 14 节 参考文献

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

第 15 节 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>

```

检查 expl3 和 l3keys2e 的版本。

```

9 <*class>
10 \msg_new:nmmm { ctex } { l3-too-old }
11 { Support~package~`#1'~too~old. }
12 {
13   Please~update~an~up~to~date~version~of~the~bundles\\\
14   `l3kernel'~and~`l3packages'\\\
15   using~your~TeX~package~manager~or~from~CTAN.
16 }
17 \ifpackagelater { expl3 } { 2014/07/20 } { }
18 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
19 \ifpackagelater { l3keys2e } { 2014/05/05 } { }
20 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
21 </class>
22 <*class|style>
23 \RequirePackage { etoolbox , ifpdf , fix-cm , everyysel }

```

15.1 内部函数与变量

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_dim
\l__ctex_tmp_box
24 \tl_new:N \l__ctex_tmp_tl
25 \int_new:N \l__ctex_tmp_int
26 \dim_new:N \l__ctex_tmp_dim
27 \box_new:N \l__ctex_tmp_box

```

(End definition for \l__ctex_tmp_tl and others.)

对旧版本的宏包给出错误信息。

```

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

```

\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。

```

34 \cs_new_eq:NN \ctex_lua_now_x:n \luatex_directlua:D

```

(End definition for \ctex_lua_now_x:n.)

\ctex_if_pdfmode:TF \ifpdf 的简单 wrapper。

```

35 \ifpdf
36 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_i:nn
37 \else:
38 \cs_new_eq:NN \ctex_if_pdfmode:TF \use_ii:nn
39 \fi:

```

(End definition for \ctex_if_pdfmode:TF.)

`\ctex_if_preamble:TF` 测试是否在 $\text{\LaTeX}2_{\epsilon}$ 的导言区。 $\text{\LaTeX}2_{\epsilon}$ 中的 `\@onlypreamble` 命令可以处理宏参数,使其在 document 环境后被重定义为 `\@notprerr`,而又由于 `\@onlypreamble` 本身也被这样处理过,因此可以测试 `\@onlypreamble` 是否与 `\@notprerr` 相同来确定是否在导言区。

```

40 \prg_new_conditional:Npnn \ctex_if_preamble: { TF }
41 {
42   \if_meaning:w \@onlypreamble \@notprerr
43   \prg_return_false:
44   \else:
45     \prg_return_true:
46   \fi:
47 }

```

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

`\ctex_at_end_package:nn` 与 `filehook` 的 `\AtEndOfPackageFile*` 类似,如果原来没有在载入宏包则在宏包末尾执行语句,否则立即执行。

```

48 \cs_new_protected:Npn \ctex_at_end_package:nn #1#2
49 {
50   \@ifpackageloaded { #1 }
51   { #2 }
52   { \ctex_add_hook:cn { g__ctex_at_end_ #1 _hook_tl } { #2 } }
53 }

```

(End definition for `\ctex_at_end_package:nn`.)

`\ctex_add_hook:Nn` 给钩子附加内容。

`\ctex_add_hook:cn`

```

54 \cs_new_protected:Npn \ctex_add_hook:Nn #1#2
55 {
56   \tl_if_exist:NF #1 { \tl_new:N #1 }
57   \tl_gput_right:Nn #1 { #2 }
58 }
59 \cs_generate_variant:Nn \ctex_add_hook:Nn { c }

```

(End definition for `\ctex_add_hook:Nn` and `\ctex_add_hook:cn`.)

`\ctex_package_end_hook:n` 宏包末尾钩子,只执行一次,用后清除。

`\ctex_package_end_hook:o`

```

60 \cs_new_protected_nopar:Npn \ctex_package_end_hook:n #1
61 {
62   \cs_if_exist_use:cT { g__ctex_at_end_ #1 _hook_tl }
63   { \cs_undefine:c { g__ctex_at_end_ #1 _hook_tl } }
64 }
65 \cs_generate_variant:Nn \ctex_package_end_hook:n { o }

```

(End definition for `\ctex_package_end_hook:n` and `\ctex_package_end_hook:o`.)

对 `\@popfilename` 做补丁来实现 `\ctex_at_end_package:nn` 的功能。

```

66 \tl_put_left:Nn \@popfilename
67 { \ctex_package_end_hook:o { \@currname } }

```

`\ctex_file_wrapper:nnn` 设置文件操作的 `\catcode` 环境,参数 #1 是设置,#2 是文件操作,#3 是恢复。默认关闭 $\text{\LaTeX}3$ 语法环境,并设置 @ 的 `\catcode` 为 11。

```

68 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
69 {
70   \use:x
71   {
72     \ExplSyntaxOff
73     \char_set_catcode_letter:n { 64 }
74     #1
75     \exp_not:n {#2}

```

```

76     \bool_if:NTF \l__kernel_expl_bool
77     { \ExplSyntaxOn }
78     { \ExplSyntaxOff }
79     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
80     #3
81   }
82 }

```

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

`\ctex_file_input:n` 输入文件。

```

83 \cs_new_protected_nopar:Npn \ctex_file_input:n #1
84 { \ctex_file_wrapper:nnn { } { \file_input:n {#1} } { } }

```

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

`\ctex_scheme_input:n` 输入 scheme 文件。先查找当前文档类下的 `(scheme)`, 找不到再查找一般的文件。

```

85 \cs_new_protected_nopar:Npn \ctex_scheme_input:n #1
86 {
87   \ctex_file_wrapper:nnn
88   { }
89   {
90     \tl_if_exist:NTF \c__ctex_class_tl
91     {
92       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
93       { \file_input:n { ctex-scheme- #1 .def } }
94     }
95     { \file_input:n { ctex-scheme- #1 .def } }
96   }
97   { }
98 }
99 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

(End definition for `\ctex_scheme_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_\u          % \DeclareRobustCommand\xaa[1]{...}
2 \protect_\xab_\u          % \DeclareRobustCommand\xab[1][]{...}
3 \@protected@testopt_\xac_\u\{ % \newcommand\xac[1][]{...}
4 \@testopt_\xad_\u         % \newrobustcmd\xad[1][]{...}
5 \x@protect_\1\protect_\1_\u % \DeclareRobustCommand\1[1]{...}
6 \x@protect_\2\protect_\2_\u % \DeclareRobustCommand\2[1][]{...}
7 \@protected@testopt_\3\3_\u % \newcommand\3[1][]{...}
8 \@testopt_\4_\u          % \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 `regexpatch` 宏包共用了主要函数的名字, 从而将导致用户不能使用 `regexpatch`。

```

100 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
101 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
102 \group_begin:
103 \char_set_lccode:nn { `< } { `\< }
104 \char_set_lccode:nn { `\/ } { `\/ }
105 \char_set_lccode:nn { `A } { `t }
106 \tl_map_function:nN { \A \E \S \O \P } \char_set_catcode_other:N
107 \tex_lowercase:D
108 {
109   \group_end:

```

```

110 \cs_new_protected:Npn \ctex_parse_name:NNn #1#2#3
111 {
112   \bool_if:nTF { \cs_if_exist_p:c { #3 ~ } || \cs_if_exist_p:c { /#3 } }
113   {
114     \group_begin:
115     \use:x
116     {
117       \__ctex_parse_name:nNNNnN { \token_get_replacement_spec:N #2 }
118       \exp_not:N #2 \exp_not:c { #3 ~ } \exp_not:c { /#3 } {#3}
119     } #1
120   }
121   { #1#2 }
122 }
123 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN #1#2#3#4#5#6
124 {
125   \group_end:
126   \exp_args:Nc #6
127   {
128     \str_case:nnTF {#1}
129     {
130       { \protect #3 } { }
131       { \x@protect #2 \protect #3 } { }
132     }
133     {
134       \str_if_eq_x:nnTF { \exp_not:n { /@protected@ #3 /#3 } }
135       {
136         \exp_last_unbraced:Nf \__ctex_parse_name:w
137         \token_get_replacement_spec:N #3 AESAOPA ~ < \q_stop
138       }
139       { /#5 ~ } { #5 ~ }
140     }
141     {
142       \str_case:onTF { \__ctex_parse_name:w #1 AESAOPA ~ < \q_stop }
143       {
144         { /@protected@ #2 #4 } { }
145         { /@ #4 } { }
146       }
147       { /#5 } {#5}
148     }
149   }
150 }
151 \cs_new:Npn \__ctex_parse_name:w #1 AESAOPA ~ #2 < #3 \q_stop { #1#2 }
152 }
153 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

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

```

\ctex_patch_cmd:NnnTF \ctex_preto_cmd:NnTF \ctex_appto_cmd:NnTF
\ctex_patch_cmd:NnnTF \ctex_preto_cmd:NnTF \ctex_appto_cmd:NnTF
\ctex_patch_cmd:NnnTF \ctex_preto_cmd:NnTF \ctex_appto_cmd:NnTF

```

(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 \LaTeX 3 语法。

```

157 \cs_new_protected:Npn \ctex_patch_cmd:Nnn
158 {
159   \group_begin:
160   \char_set_catcode_other:N \#
161   \__ctex_patch_cmd:Nnn
162 }
163 \cs_new_protected:Npn \__ctex_patch_cmd:Nnn #1#2#3
164 {

```



```

165 \group_end:
166 \group_begin:
167 \ExplSyntaxOff
168 \ctex_patch_cmd:NnnTF #1 {#2} {#3}
169 {
170   \cs_gset_eq:NN \__ctex_tmp:w #1
171   \group_end:
172   \cs_set_eq:NN #1 \__ctex_tmp:w
173   \cs_undefine:N \__ctex_tmp:w
174 }
175 { \group_end: \ctex_patch_failure:N #1 }
176 }
177 \cs_new_protected:Npn \ctex_patch_failure:N #1
178 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
179 \msg_new:nmn { ctex } { patch-failure }
180 {
181   Patching~command~`#1'~failed.\\
182   ctex~may~not~work~as~expected.
183 }

```

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

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

```

184 \cs_new:Npn \ctex_set_default_ccwd:Nn #1#2
185 {
186   \dim_compare:nNnTF
187     { \ctex_default_pt:n {#2} } = { \ctex_default_pt:n { #2 ~ mm } }
188     { \tl_set:Nn #1 {#2} }
189     { \tl_set:Nn #1 { #2 \ccwd } }
190 }

```

(End definition for `\ctex_set_default_ccwd:Nn`.)

`\ctex_default_pt:n` 最新版本的 `expl3` 已经不允许 `\dim_to_decimal: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`。

```

191 \cs_new:Npn \ctex_default_pt:n #1
192 {
193   \exp_after:wN \__ctex_default_pt:w
194   \dim_use:N \etex_dimexpr:D #1 pt \scan_stop: \q_stop
195 }
196 \group_begin:
197 \char_set_catcode_other:N \P
198 \char_set_catcode_other:N \T
199 \tex_lowercase:D
200 {
201   \group_end:
202   \cs_new:Npn \__ctex_default_pt:w #1 PT #2 \q_stop { #1 PT }
203 }

```

(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。

```

204 \tl_new:N \l__ctex_encoding_tl
205 \tl_set:Nx \l__ctex_encoding_tl
206 { \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。

```

207 \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。

```
208 \bool_new:N \g__ctex_zhmCJK_bool
```

(End definition for `\g__ctex_zhmCJK_bool`.)

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

```
209 \tl_new:N \l__ctex_autoindent_tl
```

(End definition for `\l__ctex_autoindent_tl`.)

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

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

(End definition for `\ctex_if_autoindent_touched:F`.)

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

```
211 \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。

```
212 <class>\cs_new_protected_nopar:Npn \ctex_at_end:n { \AtEndOfClass }
```

```
213 <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。

```
214 <*class>
```

```
215 \clist_new:N \g__ctex_std_options_clist
```

```
216 </class>
```

(End definition for `\g__ctex_std_options_clist`.)

对无效选项给出警告。

```
217 \msg_new:nnn { ctex } { invalid-option }
```

```
218 { Option~`\l_keys_key_tl'~is~invalid~in~current~mode. }
```

```
219 \msg_new:nnn { ctex } { invalid-value }
```

```
220 { Value~`#1'~is~invalid~for~the~key~`\l_keys_key_tl'. }
```

对过时选项或命令给出警告。

```
221 \msg_new:nnn { ctex } { deprecated-option }
```

```
222 { Option~`\l_keys_key_tl'~is~deprecated.\\ #1 }
```

```
223 \msg_new:nnn { ctex } { deprecated-command }
```

```
224 { Command~ #1 is~ deprecated.\\ #2 }
```

```
225 \msg_new:nnn { ctex } { deprecated-environment }
```

```
226 { Environment~ `#1'~is~ deprecated.\\ #2 }
```

```
227 </class|style>
```

```
228 <*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。

```
229 \cs_new_eq:NN \g__ctex_font_size_flag \c_minus_one
```

(End definition for `\g__ctex_font_size_flag`.)

15.2 宏包选项

```
230 \keys_define:nn { ctex / option }
231 {
```

zihao

```
232   zihao .choice: ,
233   zihao .default:n = { 5 } ,
234   zihao / 5 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero } ,
235   zihao / -4 .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_one } ,
236   zihao / false .code:n = { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
237 <ctexsize> }
238 </class|style|ctexsize>
239 <*class|style>
240   c5size .code:n =
241   {
242     \msg_warning:nnn { ctex } { deprecated-option }
243     { Option~ `zihao=5'~ is~ set. }
244     \keys_set:nn { ctex / option } { zihao = 5 }
245   } ,
246   cs4size .code:n =
247   {
248     \msg_warning:nnn { ctex } { deprecated-option }
249     { Option~ `zihao=-4'~ is~ set. }
250     \keys_set:nn { ctex / option } { zihao = -4 }
251   } ,
252   c5size .value_forbidden: ,
253   cs4size .value_forbidden: ,
```

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

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

```
254   linespread .fp_set:N = \l__ctex_line_spread_fp ,
255   linespread .initial:n = { \c_nan_fp } ,
256   linespread .value_required: ,
```

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

autoindent 自动调整段落的首行缩进功能。

```
257   autoindent .choice: ,
258   autoindent .default:n = { true } ,
259   autoindent / true .code:n =
260   {
261     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
262     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
263   } ,
264   autoindent / false .code:n =
265   {
266     \tl_clear:N \l__ctex_autoindent_tl
267     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
268   } ,
269   autoindent / unknown .code:n =
270   {
271     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
272     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
273   } ,
```

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

indent 仅为兼容性保留, 已过时。

```
274   indent .code:n =
275   {
276     \msg_warning:nnn { ctex } { deprecated-option }
```

```

277     {
278     The~ functionality~ has~ been~ removed.~
279     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
280     options.~
281     }
282   },
283   indent .value_forbidden: ,
284   noindent .code:n =
285   {
286     \msg_warning:nnn { ctex } { deprecated-option }
287     {
288     The~ functionality~ has~ been~ removed.~
289     It's~ better~ to~ set~ the~ heading~ styles~ via~ beforeskip~
290     options.
291     }
292   },
293   noindent .value_forbidden: ,

```

(End definition for `indent`. This function is documented on page 26.)

GBK

UTF8

```

294   GBK .code:n = { \tl_set:Nn \l__ctex_encoding_tl { GBK } } ,
295   UTF8 .code:n = { \tl_set:Nn \l__ctex_encoding_tl { UTF8 } } ,
296   GBK .value_forbidden: ,
297   UTF8 .value_forbidden: ,

```

(End definition for `GBK` and `UTF8`. These functions are documented on page 7.)

`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。

```

298   fontset .tl_gset:N = \g__ctex_fontset_tl ,
299   nofonts .code:n =
300   {
301     \msg_warning:nnn { ctex } { deprecated-option }
302     {
303     Option~ `fontset=none'~ is~ set.~ It~ is~ better~ to~ use~
304     fontset~ option.
305     }
306     \keys_set:nn { ctex / option } { fontset = none }
307   },
308   adobe-fonts .code:n =
309   {
310     \msg_warning:nnn { ctex } { deprecated-option }
311     {
312     Option~ `fontset=adobe'~ is~ set.~ It~ is~ better~ to~ use~
313     fontset~ option.
314     }
315     \keys_set:nn { ctex / option } { fontset = none }
316   },
317   winfonts .code:n =
318   {
319     \msg_warning:nnn { ctex } { deprecated-option }
320     {
321     Option~ `fontset=windows'~ is~ set.~ It~ is~ better~ to~ use~
322     fontset~ option.
323     }
324     \keys_set:nn { ctex / option } { fontset = none }
325   },
326   nofonts .value_forbidden: ,
327   winfonts .value_forbidden: ,
328   adobe-fonts .value_forbidden: ,

```

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

zhmap

```

329     zhmap .choice: ,
330     zhmap .default:n = { true } ,
331     zhmap / zhmCJK .code:n =
332     {
333         \bool_gset_true:N \g__ctex_zhmCJK_bool
334         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
335     } ,
336     zhmap / true .code:n =
337     {
338         \bool_gset_false:N \g__ctex_zhmCJK_bool
339         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
340     } ,
341     zhmap / false .code:n =
342     {
343         \bool_gset_false:N \g__ctex_zhmCJK_bool
344         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
345     } ,
346     nozhmap .code:n =
347     {
348         \msg_warning:nnn { ctex } { deprecated-option }
349         { Option~ `zhmap=false'~ is~ set. }
350         \keys_set:nn { ctex / option } { zhmap = false }
351     } ,
352     nozhmap .value_forbidden: ,

```

(End definition for zhmap. This function is documented on page 27.)

punct 设置标点符号输出格式。

```

353     punct .tl_set:N = \l__ctex_punct_tl ,
354     punct .default:n = { quanjiao } ,
355     punct .initial:n = { quanjiao } ,
356     nopunct .code:n =
357     {
358         \msg_warning:nnn { ctex } { deprecated-option }
359         { Option~ `punct=plain'~ is~ set. }
360         \keys_set:nn { ctex / option } { punct = plain }
361     } ,
362     nopunct .value_forbidden: ,

```

(End definition for punct. This function is documented on page 27.)

space

```

363     space .choices:nn =
364     { true , auto , false }
365     {
366         \exp_args:Nx \ctex_at_end:n
367         { \keys_set:nn { ctex } { space = \l_keys_choice_tl } }
368     } ,
369     space .default:n = { true } ,
370     nospace .code:n =
371     {
372         \msg_warning:nnn { ctex } { deprecated-option }
373         { Option~ `space=false'~ is~ set. }
374         \keys_set:nn { ctex / option } { space = false }
375     } ,
376     nospace .value_forbidden: ,

```

(End definition for space. This function is documented on page 27.)

heading

```

377     heading .bool_set:N = \l__ctex_heading_bool ,

```

(End definition for heading. This function is documented on page 10.)

scheme

```

378 scheme .tl_set:N = \l__ctex_scheme_tl ,
379 scheme .default:n = { chinese },
380 scheme .initial:n = { chinese },
381 cap .code:n =
382 {
383   \msg_warning:nnn { ctex } { deprecated-option }
384   { Option~ `scheme = chinese' ~ is~ set. }
385   \keys_set:nn { ctex / option } { scheme = chinese }
386 } ,
387 nocap .code:n =
388 {
389   \msg_warning:nnn { ctex } { deprecated-option }
390   { Option~ `scheme = plain' ~ is~ set. }
391   \keys_set:nn { ctex / option } { scheme = plain }
392 } ,
393 cap .value_forbidden: ,
394 nocap .value_forbidden: ,

```

(End definition for scheme. This function is documented on page 10.)

sub3section

sub4section

```

395 sub3section .code:n =
396   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_three } ,
397 sub4section .code:n =
398   { \cs_gset_eq:NN \g__ctex_section_depth_flag \c_four } ,
399 sub3section .value_forbidden: ,
400 sub4section .value_forbidden: ,

```

(End definition for sub3section and sub4section. These functions are documented on page 10.)

fntef

fancyhdr

hyperref

```

401 fntef .code:n =
402 {
403   \msg_warning:nnn { ctex } { deprecated-option }
404   { `(xe)CJKfntef'~ package~ is~ always~ loaded. }
405 } ,
406 fancyhdr .code:n =
407 {
408   \msg_warning:nnn { ctex } { deprecated-option }
409   { `fancyhdr'~ package~ is~ loaded. }
410   \RequirePackage{ fancyhdr }
411 } ,
412 hyperref .code:n =
413 {
414   \msg_warning:nnn { ctex } { deprecated-option }
415   { `hyperref'~ package~ will~ be~ loaded. }
416   \AtEndPreamble { \RequirePackage { hyperref } }
417 } ,
418 }

```

(End definition for fntef, fancyhdr, and hyperref. These functions are documented on page 27.)

```

419 </class|style>
420 <*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

```

421 \tl_clear:N \l__ctex_tmp_tl
422 \clist_map_inline:nn

```

```

423 {
424     10pt , 11pt , 12pt ,
425     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
426 }
427 {
428     \tl_put_right:Nn \l__ctex_tmp_tl
429     {
430         #1 .code:n =
431 <!*class>
432         { \cs_gset_eq:NN \g__ctex_font_size_flag \c_two } ,
433 </!class>
434 <*class>
435         {
436             \cs_gset_eq:NN \g__ctex_font_size_flag \c_two
437             \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
438         } ,
439 </class>
440         #1 .value_forbidden: ,
441     }
442 }
443 \use:x { \keys_define:nn { ctex / option } { \exp_not:o { \l__ctex_tmp_tl } } }
444 \tl_clear:N \l__ctex_tmp_tl

```

(End definition for 10pt, 11pt, and 12pt. These functions are documented on page 9.)

将未知选项传给标准文档类。

```

445 <*class>
446 \keys_define:nn { ctex / option }
447 {
448     unknown .code:n =
449         { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
450 }
451 </class>
452 <|ctexsize>\ctex_file_input:n { ctexopts.cfg }
453 \ProcessKeysOptions { ctex / option }
454 </class|style|ctexsize>
455 <*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。

```

456 <*class>
457 \if_case:w \g__ctex_font_size_flag
458     \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt }
459 \or:
460     \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt }
461 \fi:

```

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

```

462 <*article>
463 \tl_const:Nn \c__ctex_class_tl { article }
464 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
465 \LoadClass { article }
466 </article>
467 <*book>
468 \tl_const:Nn \c__ctex_class_tl { book }
469 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
470 \LoadClass { book }
471 </book>
472 <*report>
473 \tl_const:Nn \c__ctex_class_tl { report }
474 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
475 \LoadClass { report }
476 </report>
477 </class>

```

```

478 \tl_set_eq:Nc \l__ctex_tmp_tl { ver@ \@currname . \@currentx }
479 <*class>
480 \cs_new_eq:cN { ver@ctex. \@pkgextension } \l__ctex_tmp_tl
481 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \l__ctex_tmp_tl
482 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl
483 </class>
484 <*style>
485 \msg_new:nnnn { ctex } { ctexsize-loaded }
486 { Package~`ctexsize'~can~not~be~loaded~before~`ctex'. }
487 {
488   `ctexsize'~is~actually~a~part~of~`ctex'.\\
489   It~is~not~necessary~to~load~it~separately.
490 }
491 \ifpackageloaded { ctexsize }
492 { \msg_error:nn { ctex } { ctexsize-loaded } }
493 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \l__ctex_tmp_tl }
494 </style>

```

15.3 用户设置接口

`\ctexset`

```

495 \NewDocumentCommand \ctexset { +m }
496 { \keys_set:nn { ctex } {#1} }

```

(End definition for `\ctexset`. This function is documented on page 6.)

`\CTEXsetup` `\CTEXoptions` 过时命令。

```

497 \NewDocumentCommand \CTEXsetup { +o > { \TrimSpaces } m }
498 {
499   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXsetup }
500   { \ctexset~ {~ #2~ =~ {~ #1~ }~ }~ is~ set. }
501   \IfNoValueF {#1} { \keys_set:nn { ctex / #2 } {#1} }
502 }
503 \NewDocumentCommand \CTEXoptions { +o }
504 {
505   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXoptions }
506   { \ctexset~ {~ #1~ }~ is~ set. }
507   \IfNoValueF {#1} { \keys_set:nn { ctex } {#1} }
508 }

```

(End definition for `\CTEXsetup` and `\CTEXoptions`. These functions are documented on page 27.)

```

509 </class|style>

```

15.4 特定引擎支持与设置

15.4.1 `ctex-engine-pdftex.def`

```

510 <*pdftex>

```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```

511 \if_bool:N \g__ctex_zhmCJK_bool
512 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhmCJK }
513 \RequirePackage { zhmCJK }

```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```

514 \else:
515 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
516 { \RequirePackage { CJK } }
517 { \RequirePackage { CJKutf8 } }
518 \RequirePackage { CJKpunct , CJKspace }

```


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

```

519 \cs_new_protected_nopar:Npn \ctex_load_zhmap:n #1#2#3#4
520   {
521     \tl_set:Nn \CJKrmdefault {#1}
522     \tl_set:Nn \CJKsfdefault {#2}
523     \tl_set:Nn \CJKttdefault {#3}
524     \AtBeginDvi { \file_input:n {#4} }
525     \ctex_at_end_package:nn { atbegshi }
526     { \AtBeginShipoutFirst { \file_input:n {#4} } }
527   }
528 \@onlypreamble \ctex_load_zhmap:n

(End definition for \ctex_load_zhmap:n)

529 \tl_if_exist:NF \CJKfamilydefault
530   { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
531 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
532 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
533 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
534 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
535   { \ctex_patch_failure:N \rmfamily }
536 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
537   { \ctex_patch_failure:N \sffamily }
538 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
539   { \ctex_patch_failure:N \ttfamily }
540 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
541   { \cs_set_eq:NN \reset@font \normalfont }
542   { \ctex_patch_failure:N \normalfont }

```

`zhmCJK` 判断结束。

```
543 \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。

```

544 \cs_new_protected_nopar:Npn \ctex_CJK_input:n #1
545   {
546     \ctex_file_wrapper:nnn
547     {
548       \char_set_catcode_other:n { 60 } % <
549       \char_set_catcode_math_superscript:n { 94 } % ^
550       \int_set_eq:NN \tex_endlinechar:D \c_minus_one
551     }
552     { \file_input:n {#1} }
553     {
554       \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
555       \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
556       \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
557     }
558   }
559 \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` `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¹⁴。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

560 \if_cs_exist:N \CJK@surr
561 \cs_new_protected_nopar:Npn \ctex_plane_to_utfxvibe:Nn #1#2
562   {

```

¹⁴<https://github.com/CTeX-org/ctex-kit/issues/146>

```

563 \tl_set:Nx \l__ctex_tmp_tl {#2}
564 \int_set:Nn \l__ctex_tmp_int
565   { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
566 \int_compare:nNnTF \l__ctex_tmp_int < \c_two_hundred_fifty_six
567   { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
568   {
569     \int_sub:Nn \l__ctex_tmp_int { \c_two_hundred_fifty_six }
570     \tl_gset:Nx #1
571     {
572       \int_to_Hex:n
573       { \int_div_truncate:nn { \l__ctex_tmp_int } { \c_four } + "D800 }
574       \int_to_Hex:n
575       { \int_mod:nn { \l__ctex_tmp_int } { \c_four } + "DC }
576     }
577   }
578 }
579 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
580 \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`。

```

581 \AtBeginDocument
582 {
583   \str_if_eq:x:nnF { \l__ctex_punct_tl } { quanjiao }
584   { \punctstyle { \l__ctex_punct_tl } }
585 }

```

启用中文字符功能。GBK 编码时, 将汉字的首字节设置为活动字符, 并对这些字符初始化; UTF8 编码时, 上游宏包已经处理好。 `\CJK@makeActive` 应该先于 `ctex-name-gbk.cfg` 等文件的载入。注意 `\CJK@loadBinding` 需要调用补丁后的 `\CJK@input`。使用 `zhmCJK` 时, 此功能已经被启用。

```

586 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
587 \str_if_eq:onF { \l__ctex_encoding_tl } { UTF8 }
588 {
589   \CJK@makeActive
590   \CJK@loadBinding { standard }
591 }

```

在导言区结束时调用 `\CJK@envStart` 启用完整的中文功能。

`\CJK@envStart` 的定义是

```

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

```

`\CJK@upperReset` 可能会有一定风险, 因此我们直到导言区末尾才使用 `\CJK@envStart`。

这样可以避免将 CJK 环境内置入 document 环境的最里层，最后也就不需要 `\clearpage`。`zhmCJK` 已经提供类似功能。

```
592 \exp_args:Nx \AtEndPreamble
593 {
594   \exp_not:N \CJK@envStart
595   { } { \l__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
596   \exp_not:N \CJKtilde
597 }
```

`zhmCJK` 判断结束。

```
598 \fi:
```

`\ctex_auto_ignorespaces`: 关闭名字空间,保存 `\CJK@@ignorespaces` 的定义,方便使用。

```
599 <@@=>
600 \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。

```
601 <@@=ctex>
```

(End definition for \ctex_auto_ignorespaces:.)

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

```
602 \cs_new_protected_nopar:Npn \ctex_ignorespaces_case:N #1
603 {
604   \cs_set_protected_nopar:Npn \ctex_set_ignorespaces:
605   { \cs_set_eq:NN \CJK@ignorespaces #1 }
606   \ctex_set_ignorespaces:
607 }
608 \cs_new_protected_nopar:Npn \ctex_set_ignorespaces:
609 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }
```

(End definition for \ctex_ignorespaces_case:N and \ctex_set_ignorespaces:.)

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

```
610 \ctex_add_hook:Nn \CJKhook { \ctex_set_ignorespaces: }
```

(End definition for \CJKhook.)

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

```
611 \cs_new_protected_nopar:Npn \ctex_punct_set:n #1
612 {
613   \clist_map_inline:Nn \c__ctex_punct_family_clist
614   {
615     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
616     {
617       \cs_set_eq:cc
618       { CJKpunct@ ##1 @spaces }
619       { c__ctex_ #1 ##1 _punct_spaces_tl }
620     }
621   }
622 }
623 \clist_const:Nn \c__ctex_punct_family_clist
624 {
625   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
626   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
627 }
```

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

```
\ctex_punct_map_family:nn CJK 族 #1 使用族 #2 的边界信息。
628 \cs_new_protected_nopar:Npn \ctex_punct_map_family:nn #1#2
629 {
630   \cs_if_free:cF { CJKpunct@ #2 @spaces }
631   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
632 }
```

(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。
633 \cs_new_protected_nopar:Npn \ctex_punct_map_bfseries:nn #1#2
634 {
635   \clist_map_inline:nn {#1}
636   {
637     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
638     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
639   }
640 }
641 \cs_new_protected_nopar:Npn \ctex_punct_map_series:nnn #1#2#3
642 {
643   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
644   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
645   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
646   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
647   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
648   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
649 }
```

(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。
650 \cs_new_protected_nopar:Npn \ctex_punct_map_itshape:nn #1#2
651 {
652   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
653   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
654   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
655   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
656   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
657   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
658 }
```

(End definition for `\ctex_punct_map_itshape:nn`.)

```
\ctex_punct_space:nn 定义标点的边界信息。
\ctexspadef
659 \cs_new_protected_nopar:Npn \ctex_punct_space:nn #1#2
660 { \tl_const:cn { c_#1_ctex_#1_punct_spaces_tl } {#2} }
661 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn
```

(End definition for `\ctex_punct_space:nn` and `\ctexspadef`.)

载入边界信息文件。

```
662 \ctex_file_input:n { ctexspa.def }
663 </pdfTeX>
```

15.4.2 ctex-engine-xetex.def

```
664 < *xetex >
665 \RequirePackage { xeCJK }
```

```

666 \exp_args:Nx \xeCJKsetup
667 {
668   LoadFandol = false ,
669   AutoFakeBold = true ,
670   PunctStyle = \l__ctex_punct_tl
671 }

```

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`。

```

672 \@ifpackagelater { fontspec } { 2014/05/25 } { }
673 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
674 </xetex>

```

15.4.3 ctex-engine-luatex.def

```

675 <*luatex>

```

LuaTeX-ja 为了兼容 pL^AT_EX 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。我们在这里禁用它。

```

676 \msg_new:nnn { ctex } { luatexja-loaded }
677 {
678   Package~luatexja'~can~not~be~loaded~before~`ctex'.\\
679   Loading~file~`#1'~will~abort!
680 }
681 \@ifpackageloaded { luatexja }
682 { \msg_critical:nxx { ctex } { luatexja-loaded } { \g_file_current_name_tl } }
683 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
684 \RequirePackage { luatexja }
685 \@ifpackagelater { luatexja-core } { 2015/02/22 } { }
686 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
687 \RequirePackage { fontspec }
688 \@ifpackagelater { fontspec } { 2014/05/25 } { }
689 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

15.4.3.1 LuaTeX-ja 的默认设置

```

690 \ExplSyntaxOff

```

以下设置抄录自 `lltjdefs.sty`。

```

691 \ltjdefcharrange{1}{"80-"36F, "1E00-"1EFF}
692 \ltjdefcharrange{2}{"370-"4FF, "1F00-"1FFF}
693 \ltjdefcharrange{3}{%
694   "2000-"243F, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
695 \ltjdefcharrange{4}{%
696   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
697   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A82F, "A840-"ABFF, "FB00-"FE0F,
698   "FE20-"FE2F, "FE70-"FEFF, "10000-"1FFFF, "E000-"F8FF} % non-Japanese
699 \ltjdefcharrange{5}{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
700 \ltjdefcharrange{6}{%
701   "2460-"24FF, "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
702   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE6F, "20000-"2FFFF, "E0100-"E01EF}
703 \ltjdefcharrange{7}{
704   "1100-"11FF, "2F00-"2FFF, "3100-"31EF, "A000-"A4CF, "A830-"A83F,
705   "AC00-"D7FF}
706 \ltjdefcharrange{8}{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}
707 \ltjsetparameter{jacharrange={-1, +2, +3, -4, -5, +6, +7, +8}}
708 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 `ltj-latex.sty`。

```

709 \directlua{
710   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
711   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
712 }
713 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .4pt,

```

```

714 xkanjiskip=.25\zw plus 1pt minus 1pt,
715 autospacing, autoxspacing, jacharrange={-1},
716 yalbaselineshift=\z@, yjabaselineshift=\z@,
717 jcharwidowpenalty=500, differentjfm=paverage
718 }
719 \ExplSyntaxOn

```

15.4.3.2 LuaTeX-ja 的补丁

```
720 <@@=ctex_ltj>
```

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

```

721 \RequirePackage { xunicode-addon }
722 \AtBeginUTFCommand
723 {
724   \group_begin:
725   \ctex_lua_now_x:n { tex.globaldefs = 0 }
726   \ltj@allalchar
727 }
728 \AtEndUTFCommand { \group_end: }

```

对 fontspec 沿用 LuaTeX-ja 的补丁。

```
729 \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。

```

730 \group_begin:
731 \char_set_catcode_other:n { \c_zero }
732 \cs_new_protected:Npn \__ctex_ltj_um_char:Nw #1 = #2 \q_nil
733 {
734   \group_begin:
735   \char_set_lccode:nn { \c_zero } {#2}
736   \tex_lowercase:D
737   {
738     \group_end:
739     \cs_gset_protected_nopar:Npn #1
740     {
741       \mode_if_math:TF { ^^@ }
742       { {
743         \ctex_lua_now_x:n { tex.globaldefs = 0 }
744         \ltj@allalchar ^^@
745       } }
746     }
747   }
748   \ltjsetmathletter {#2}
749 }
750 \group_end:
751 \ctex_at_end_package:nn { unicode-math }
752 {
753   \cs_if_free:NF \um_cs_set_eq_active_char:Nw
754   { \cs_set_eq:NN \um_cs_set_eq_active_char:Nw \__ctex_ltj_um_char:Nw }
755 }

```

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

```

756 \ctex_at_end_package:nn { listings }
757 {
758   \use:x
759   {
760     \exp_not:N \RequirePackage { lltjp-listings }
761     \tl_set:Nn \exp_not:N \lstlistingname
762     { \exp_not:o { \lstlistingname } }

```

```

763     \tl_set:Nn \exp_not:N \lstlistlistingname
764     { \exp_not:o { \lstlistlistingname } }
765   }
766 }

```

15.4.3.3 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
767 \cs_new_protected_nopar:Npn \ctex_ltj_select_font:
768 {
769   \cs_if_exist_use:cF { \l__ctex_ltj_current_font_tl }
770   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
771 }
772 \tl_new:N \CJK@family
773 \tl_new:N \l__ctex_ltj_current_font_tl
774 \tl_set:Nn \l__ctex_ltj_current_font_tl
775 { \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。

```

776 \cs_new_protected_nopar:Npn \__ctex_ltj_select_font_aux:
777 {
778   \group_begin:
779   \tl_set_eq:NN \f@encoding \CJK@encoding
780   \tl_set_eq:NN \f@family \CJK@family
781   \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
782   \ctex_ltj_pickup_font:
783   \group_end:
784   \font@name
785   \__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。

```

786   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
787   { \__ctex_ltj_select_font_aux: }
788 }
789 \cs_new_protected_nopar:Npn \__ctex_ltj_push_fontname:n #1
790 {
791   \cs_gset_eq:NN \__ctex_ltj_save_fontname:w \font@name
792   \cs_gset_nopar:Npx \font@name {#1}
793 }
794 \cs_new_protected_nopar:Npn \__ctex_ltj_pop_fontname:
795 { \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`。

```

796 \cs_new_protected_nopar:Npn \ctex_ltj_pickup_font:
797 {
798   \exp_after:wN \cs_if_exist:NF \font@name
799   {
800     \group_begin:
801     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
802     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
803     \define@newfont
804     \group_end:
805   }
806 }
807 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

(End definition for `\ctex_ltj_pickup_font:`)

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

```

%%%%\jfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont{%
  \afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(false, 'yoko')}}
\protected\def\globaljfont{%
  \afterassignment\ltj@jfont\directlua{luatexja.jfont.jfontdefX(true, 'yoko')}}
\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 ',
  cstemp, '\\endcsname{\\ltj@cur'..
  (jfm_dir == 'yoko' and 'j' or 't') .. 'fnt', fn, '\\relax}')

```

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

```

808 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
809 {
810   \get@external@font
811   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
812   {
813     \tl_set:Nx \external@font
814     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
815   }
816   \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。

```

817   \font@name
818   \ctex_lua_now_x:n { font.current(tex.attribute['ltj@curjfont']) }
819   \use:c { \f@encoding + \f@family }
820   \use:c { \curr@fontshape }
821 }

```

(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。

```

822 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
823 {
824   \ctex_ltj_if_alternate_shape_exist:nF { \curr@fontshape }
825   {
826     \group_begin:
827     \tl_set_eq:NN \CJK@family \f@family
828     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
829     {

```



```

830         \cs_gset_protected_nopar:Npx \subst@correction
831         {
832             \cs_new_eq:NN
833             \exp_not:c { \l__ctex_ltj_current_font_tl }
834             \font@name
835         }
836         \group_insert_after:N \group_insert_after:N
837         \group_insert_after:N \subst@correction
838     }
839     \group_end:
840 }
841 }

```

(End definition for `\ctex_ltj_subst_font:.`)

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

```

842 \prg_new_conditional:Npnn \ctex_ltj_if_alterate_shape_exist:n #1 { T , F , TF }
843 {
844     \ctex_lua_now_x:n { luatexja.jfont.does_alt_set (' \luatexluaescapestring {#1}' ) }
845     \prg_return_true: \else: \prg_return_false: \fi:
846 }

```

(End definition for `\ctex_ltj_if_alterate_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。

```

847 \cs_new_nopar:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
848 { #1 \ctex_lua_now_x:n { luatexja.jfont.print_aftl_address() } ~ at }

```

(End definition for `__ctex_ltj_patch_external_font:w`.)

`\ctex_ltj_select_alterate_font:` 在 `\selectfont` 中更新替代字体。

```

849 \cs_new_protected_nopar:Npn \ctex_ltj_select_alterate_font:
850 {
851     \ctex_ltj_if_alterate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
852     {
853         \ctex_lua_now_x:n
854         {
855             luatexja.jfont.output_alt_font_cmd
856             ('y', '\luatexluaescapestring { \l__ctex_ltj_current_shape_tl }')
857         }
858         \ctex_lua_now_x:n { luatexja.jfont.pickup_alt_font_a (' \f@size' ) }
859     }
860 }
861 \tl_new:N \l__ctex_ltj_current_shape_tl
862 \tl_set:Nn \l__ctex_ltj_current_shape_tl
863 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

(End definition for `\ctex_ltj_select_alterate_font:.`)

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

```

864 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@auxy #1
865 {
866     \cs_if_exist:cF { #1/\f@size }
867     {

```

```

868     \group_begin:
869     \use:x { \exp_not:N \split@name #1 / \f@size } \@nil
870     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
871     \ctex_ltj_pickup_font:
872     \group_end:
873     \__ctex_ltj_pop_fontname:
874   }
875 }

```

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

```
876 <@@=>
```

\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。

```

877 \cs_new_protected_nopar:Npn \ltj@pickup@altfont@copy #1#2
878 {
879   \ltj@@getjfontnumber #1
880   \ctex_lua_now_x:n
881   {
882     luatexja.jfont.pickup_alt_font_b
883     ( \the\ltj@tempcntc, '\luatexluaescapestring {#2}' )
884   }
885 }

```

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

```
886 \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。

```

887 \def\ltj@@IsFontJapanese#1{%
888   \directlua{luatexja.jfont.is_kenc(string.match(
889     '\luatexluaescapestring{#1}', '[^/+]'))}
890   {\catcode`M=12%
891     \gdef\ltj@@mathJapaneseFonts#1M#2#3\relax{\ltj@@IsFontJapanese{#3}}
892     \let\ltj@@al@getanddefine@fonts=\getanddefine@fonts
893     \def\ltj@@ja@getanddefine@fonts#1#2{%
894       \xdef\font@name{\csname \string#2/\tf@size\endcsname}%
895       \pickup@jfont\let\textfont@name\font@name
896       \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
897       \pickup@jfont\let\scriptfont@name\font@name
898       \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
899       \pickup@jfont
900       \edef\math@fonts{\math@fonts\ltj@setpar@global%
901         \ltj@@set@stackfont#1,\textfont@name:{MJT}%
902         \ltj@@set@stackfont#1,\scriptfont@name:{MJS}%
903         \ltj@@set@stackfont#1,\font@name:{MJSS}%
904       }%
905     }
906     \def\getanddefine@fonts#1#2{%
907       \ltj@tempcnta=#1\ltj@@IsFontJapanese{\string#2}%
908       \ifin@let\ltj@temp=\ltj@@ja@getanddefine@fonts%
909       \else \let\ltj@temp=\ltj@@al@getanddefine@fonts\fi
910       \ltj@temp{#1}{#2}%
911     }
912     \def\use@mathgroup#1#2{\relax\ifmmode
913       \math@bgroup
914       \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
915       #1\fi\ltj@tempcnta=#2 \expandafter\ltj@@mathJapaneseFonts\string#1\relax%
916       \ifin@\jfam#2\relax\else\mathgroup#2\relax\fi
917       \expandafter\math@egroup\fi}%
918     \let\@@italiccorr=\/
919   <@@=ctex_ltj)

```

```
920 \ExplSyntaxOn
```

```
\ctex_mono_jfm:n LuaTeX-ja 中与标点格式 plain 对应的 JFM 是 mono。
\l__ctex_ltj_jfm_tl
921 \cs_new_protected_nopar:Npn \ctex_mono_jfm:n #1
922 {
923   \str_if_eq:nnTF {#1} { plain }
924     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
925     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
926 }
927 \tl_new:N \l__ctex_ltj_jfm_tl
928 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
929 \ctex_mono_jfm:o { \l__ctex_punct_tl }
```

(End definition for \ctex_mono_jfm:n and \l__ctex_ltj_jfm_tl.)

\CJK@encoding 在 L^AT_EX 下, LuaTeX-ja 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:
```

```
930 \tl_const:Nn \CJK@encoding { LTJY3 }
931 \DeclareFontEncoding { \CJK@encoding } { } { }
932 \DeclareFontSubstitution { LTJY3 } { song } { \mddefault } { \updefault }
933 \ctex_lua_now_x:n { luatexja.jfont.add_kyenc_list('\CJK@encoding') }
934 \cs_new_protected_nopar:Npn \_ctex_ltj_change_encoding:
935   { \tl_set_eq:NN \g_fontspec_encoding_tl \CJK@encoding }
936 \DeclareFontFamily { \CJK@encoding } { song } { }
937 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \updefault }
938   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\_ctex_ltj_jfm_tl } { }
939 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \updefault }
940   { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\_ctex_ltj_jfm_tl } { }
941 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
942 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
943   { \CJK@encoding } { song } { \mddefault } { \updefault }
944 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
945   { \CJK@encoding } { song } { \bfdefault } { \updefault }
946 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
947 \jfam \c__ctex_ltj_math_fam_int
```

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

15.4.3.4 字体族的定义与使用

这是 luatexja-fontspec 中新增的一些字体选项。

```
948 \newfontfeature { CID } { cid = #1 }
949 \newfontfeature { JFM } { jfm = #1 }
950 \newfontfeature { JFM-var } { jfmvar = #1 }
```

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

```
951 \keys_define:nn { fontspec-preparse-external }
952 {
953   NoEmbed .code:n =
954     { \cs_set_eq:NN \_fontspec_namewrap:n \_ctex_ltj_noembed_wrap:n }
955 }
956 \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。

```
957 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
958 {
959   \group_begin:
960   \clist_clear:N \l__ctex_ltj_char_range_clist
```

```

961 \seq_clear:N \l__ctex_ltj_alternate_seq
962 \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
963 \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
964 \clist_set:N \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
965 \ctex_ltj_set_alternate_family:nnF {#1} {#3}
966 {
967   \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
968   \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
969     {#1} { \l__ctex_ltj_font_options_clist }
970   __ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
971   __ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
972   __ctex_ltj_gset_family_cs:nn {#1} {#3}
973 }
974 \group_end:
975 }
976 \cs_new_protected:Npn \ctex_ltj_set_family:xxx #1#2#3
977 { \use:x { \ctex_ltj_set_family:nnn {#1} {#2} {#3} } }
978 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
979 \clist_new:N \l__ctex_ltj_font_options_clist
980 \cs_new_protected_nopar:Npn \__ctex_ltj_use_global_options:N #1
981 {
982   \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
983   \clist_put_left:Nx #1 { JFM = \l__ctex_ltj_jfm_tl }
984 }

```

(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`

```

985 \prop_new:N \g__ctex_ltj_family_name_prop
986 \prop_new:N \g__ctex_ltj_family_font_name_prop
987 \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。

```

988 \cs_new_protected_nopar:Npn \__ctex_ltj_check_family:n #1
989 {
990   \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
991   {
992     \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
993     \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
994     \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
995     {
996       \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
997       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
998       \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
999       \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1000     }
1001     \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1002   }
1003 }
1004 \tl_new:N \l__ctex_ltj_tmp_tl
1005 \msg_new:nnn { ctex } { redefine-family }
1006 { 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`。

```

1007 \cs_new_protected_nopar:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1008 {
1009   \cs_gset_protected_nopar:cpx { \__ctex_ltj_family_csname:n {#1} }

```

```

1010     {
1011     \group_begin:
1012     \__ctex_ltj_change_encoding:
1013     \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1014     \exp_not:n { \fontspec_set_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1015     { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1016     \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1017     { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1018     \tl_gset_eq:NN \exp_not:N \g__ctex_ltj_fontspec_family_tl
1019     \exp_not:N \g__ctex_ltj_fontspec_family_tl
1020     \__ctex_ltj_set_alternate_family:n {#1}
1021     \group_end:
1022     }
1023   }
1024   \tl_new:N \l__ctex_ltj_base_family_tl
1025   \tl_new:N \g__ctex_ltj_fontspec_family_tl
1026   \cs_new_nopar:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1027   \cs_new_protected_nopar:Npn \__ctex_ltj_set_alternate_family:n #1
1028   {
1029     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1030     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1031     \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1032     \cs_if_exist_use:c { \__ctex_ltj_alternate_cs:n {#1} }
1033   }
1034   \cs_new:Npn \__ctex_ltj_alternate_cs:n #1 { ctex_ltj/alternate_family/#1 }

```

(End definition for __ctex_ltj_gset_family_cs:nn.)

\CJKfamily 切换字体。

```

1035 \NewDocumentCommand \CJKfamily { m }
1036 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1037 \cs_new_protected_nopar:Npn \ctex_ltj_switch_family:n #1
1038 {
1039   \ctex_ltj_family_if_exist:xNTF {#1} \CJK@family
1040   {
1041     \tl_set:Nn \l_ctex_ltj_family_tl {#1}
1042     \selectfont
1043   }
1044   { \__ctex_ltj_family_unknown_warning:n {#1} }
1045 }
1046 \tl_new:N \l_ctex_ltj_family_tl
1047 \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。

```

1048 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:xN #1#2 { T , F , TF }
1049 {
1050   \prop_get:NxNTF \g__ctex_ltj_family_name_prop {#1} #2
1051   { \prg_return_true: }
1052   {
1053     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1054     {
1055       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1056       \prg_return_true:
1057     }
1058     { \prg_return_false: }
1059   }
1060 }
1061 \cs_generate_variant:Nn \prop_get:NnNTF { Nx }

```

(End definition for \ctex_ltj_family_if_exist:nTF.)

__ctex_ltj_family_unknown_warning:n

```

1062 \cs_new_protected_nopar:Npn \__ctex_ltj_family_unknown_warning:n #1
1063 {
1064   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1065   {
1066     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1067     {
1068       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1069       \msg_warning:nnn { ctex } { family-unknown } {#1}
1070     }
1071   }
1072 }
1073 \seq_new:N \g__ctex_ltj_unknown_family_seq
1074 \msg_new:nnn { ctex } { family-unknown }
1075 {
1076   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1077   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1078 }
1079 \cs_new_nopar:Npn \__ctex_ltj_msg_def_family_map:n #1
1080 {
1081   \str_case_x:nnF {#1}
1082   {
1083     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1084     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1085     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1086   }
1087   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1088   [...] \{...\}
1089 }
1090 \cs_new_nopar:Npn \__ctex_ltj_msg_family_map:n #1
1091 {
1092   \str_case_x:nnF {#1}
1093   {
1094     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1095     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1096     \CJKttdefault { \token_to_str:N \CJKttdefault }
1097   }
1098   {#1}
1099 }

```

(End definition for __ctex_ltj_family_unknown_warning:n.)

\ctex_ltj_fontspec:nn

```

1100 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:nn #1#2
1101 {
1102   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1103   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1104   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1105   {
1106     \int_gincr:N \g__ctex_ltj_family_int
1107     \__ctex_ltj_fontspec:xnn
1108     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1109     {#1} {#2}
1110   }
1111 }
1112 \cs_new_protected_nopar:Npn \ctex_ltj_fontspec:xx #1#2
1113 { \use:x { \ctex_ltj_fontspec:nn {#1} {#2} } }
1114 \cs_new_protected_nopar:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1115 {
1116   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1117   {
1118     \cs_if_free:cF
1119     { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1120     {
1121       \cs_gset_eq:cc
1122       { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1123       { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }

```

```

1124         \cs_gset_eq:cc
1125         { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1126         { \__ctex_ltj_alterate_cs:n { clear / \l_ctex_ltj_family_tl } }
1127     }
1128     \bool_set_false:N \l__ctex_ltj_add_alterate_bool
1129 }
1130 \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1131 \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1132 \ctex_ltj_switch_family:n {#1}
1133 }
1134 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { x }
1135 \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
1136 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:n #1
1137 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1138 \cs_new_protected_nopar:Npn \ctex_ltj_add_font_features:nn #1#2
1139 {
1140     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1141     {#1} \l__ctex_ltj_tmp_tl
1142     {
1143         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1144         {#1} \l__ctex_ltj_font_options_clist
1145         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1146         \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1147         \ctex_ltj_fontspec:xx
1148         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1149         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1150     }
1151     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1152 }
1153 \bool_new:N \l__ctex_ltj_add_alterate_bool
1154 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1155 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1156 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1157 {
1158     \token_to_str:N \addCJKfontfeature (s)~ignored.\\
1159     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1160 }

```

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

```

\setCJKfamilyfont
\newCJKfontfamily
\CJKfontspec
\addCJKfontfeatures
1161 \NewDocumentCommand \setCJKfamilyfont { m O { } m }
1162 { \ctex_ltj_set_family:xxx {#1} {#2} {#3} }
1163 \NewDocumentCommand \newCJKfontfamily { o m O { } m }
1164 {
1165     \tl_set:Nx \l__ctex_ltj_tmp_tl
1166     { \IfNoValueTF {#1} { \cs_to_str:N #2 } {#1} }
1167     \cs_new_protected_nopar:Npx #2
1168     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1169     \ctex_ltj_set_family:xxx { \l__ctex_ltj_tmp_tl } {#3} {#4}
1170 }
1171 \NewDocumentCommand \CJKfontspec { O { } m }
1172 {
1173     \ctex_ltj_fontspec:xx {#1} {#2}
1174     \tex_ignorespaces:D
1175 }
1176 \NewDocumentCommand \addCJKfontfeatures { m }
1177 {
1178     \ctex_ltj_add_font_features:x {#1}
1179     \tex_ignorespaces:D
1180 }

```

```
1181 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures
```

(End definition for \setCJKfamilyfont and others.)

```
\setCJKmainfont
\setCJKsansfont
\setCJKmonofont
\setCJKmathfont
\defaultCJKfontfeatures
1182 \NewDocumentCommand \setCJKmainfont { 0 { } m }
1183 {
1184   \ctex_ltj_set_family:xxx { \CJKrmdefault } {#1} {#2}
1185   \normalfont
1186 }
1187 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1188 \NewDocumentCommand \setCJKsansfont { 0 { } m }
1189 {
1190   \ctex_ltj_set_family:xxx { \CJKsfdefault } {#1} {#2}
1191   \normalfont
1192 }
1193 \NewDocumentCommand \setCJKmonofont { 0 { } m }
1194 {
1195   \ctex_ltj_set_family:xxx { \CJKttdefault } {#1} {#2}
1196   \normalfont
1197 }
1198 \NewDocumentCommand \setCJKmathfont { 0 { } m }
1199 { \ctex_ltj_set_family:xxx { \c__ctex_ltj_math_tl } {#1} {#2} }
1200 \NewDocumentCommand \defaultCJKfontfeatures { m }
1201 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1202 \clist_new:N \g__ctex_ltj_default_features_clist
1203 \@onlypreamble \setCJKmainfont
1204 \@onlypreamble \setCJKsansfont
1205 \@onlypreamble \setCJKmonofont
1206 \@onlypreamble \setCJKmathfont
1207 \@onlypreamble \setCJKromanfont
1208 \@onlypreamble \defaultCJKfontfeatures
```

(End definition for \setCJKmainfont and others.)

```
1209 \tl_if_exist:NF \CJKfamilydefault
1210 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1211 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1212 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1213 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1214 \ctex_preto_cmd:NnTF \rmfamily { \CJKfamily { \CJKrmdefault } } { }
1215 { \ctex_patch_failure:N \rmfamily }
1216 \ctex_preto_cmd:NnTF \sffamily { \CJKfamily { \CJKsfdefault } } { }
1217 { \ctex_patch_failure:N \sffamily }
1218 \ctex_preto_cmd:NnTF \ttfamily { \CJKfamily { \CJKttdefault } } { }
1219 { \ctex_patch_failure:N \ttfamily }
1220 \ctex_preto_cmd:NnTF \normalfont { \CJKfamily { \CJKfamilydefault } }
1221 { \cs_set_eq:NN \reset@font \normalfont }
1222 { \ctex_patch_failure:N \normalfont }
```

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

```
1223 \cs_new_protected_nopar:Npn \ctex_ltj_ensure_default_family:
1224 {
1225   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1226   {
1227     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmpa_tl
1228     {
1229       \str_if_eq_x:nnTF { \CJKfamilydefault } { \CJKrmdefault }
1230       { \use:n }
1231       {
1232         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmpa_tl
1233         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1234         { \use:n }
1235       }
1236     }
```



```

1237         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1238         {
1239             \prop_map_break:n
1240             { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1241         }
1242     }
1243 }
1244 \normalfont
1245 \ctex_ltj_update_mathfont:
1246 }
1247 }

```

(End definition for `\ctex_ltj_ensure_default_family:.`)

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

```

1248 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:
1249 {
1250     \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1251     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1252     {
1253         \ctex_ltj_family_if_exist:xNTF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1254         { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1255     }
1256 }
1257 \cs_new_protected_nopar:Npn \ctex_ltj_update_mathfont:n #1
1258 {
1259     \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1260     \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1261     { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1262     \cs_if_free:cTF
1263     { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\updefault }
1264     {
1265         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1266         { \c__ctex_ltj_math_family_tl } { \mddefault } { \updefault }
1267     }
1268     {
1269         \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1270         { \c__ctex_ltj_math_family_tl } { \bfdefault } { \updefault }
1271     }
1272 }

```

(End definition for `\ctex_ltj_update_mathfont:.`)

15.4.3.5 替代字体的设置

AlternateFont 设置替代字体的选项。

CharRange

```

1273 \keys_define:nn { ctex_ltj / fontspec }
1274 {
1275     AlternateFont .code:n = \ctex_ltj_set_alternate_seq:n {#1} ,
1276     AlternateFont .value_required: ,
1277     CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1278     CharRange .value_required:
1279 }

```

(End definition for `AlternateFont` and `CharRange`. These functions are documented on page 23.)

`\ctex_ltj_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。

```

1280 \group_begin:
1281 \char_set_catcode_active:N \ /
1282 \char_set_lccode:nn { `\/ } { `| }
1283 \tex_lowercase:D

```

```

1284 {
1285   \group_end:
1286   \cs_new_protected:Npn \ctex_ltj_set_alternate_seq:n #1
1287   {
1288     \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1289     {
1290       \tl_set:Nn \l__ctex_ltj_tmp_tl {#1}
1291       \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { // } { || }
1292       \seq_set_split:NnV \l__ctex_ltj_tmp_seq { || } \l__ctex_ltj_tmp_tl
1293       \seq_set_filter:Nnn \l__ctex_ltj_tmp_seq \l__ctex_ltj_tmp_seq
1294       { ! \tl_if_blank_p:n { ##1 } }
1295       \seq_concat:Nnn \l__ctex_ltj_alternate_seq
1296       \l__ctex_ltj_alternate_seq \l__ctex_ltj_tmp_seq
1297     }
1298   }
1299 }
1300 \seq_new:N \l__ctex_ltj_tmp_seq
1301 \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。

```

1302 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_family:nnF #1#2#3
1303 {
1304   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1305   {
1306     \__ctex_ltj_check_family:n {#1}
1307     \seq_if_empty:NF \l__ctex_ltj_alternate_seq
1308     { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1309     #3
1310   }
1311   { \ctex_ltj_set_alternate_family:nn {#1} {#2} }
1312 }

```

(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
1313 \cs_new_protected_nopar:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1314 {
1315   \seq_map_inline:Nn \l__ctex_ltj_alternate_seq
1316   { \ctex_ltj_save_alternate_seq:Nnnwnw #1 {#2} ##1 { } \q_stop }
1317 }
1318 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }
1319 \NewDocumentCommand \ctex_ltj_save_alternate_seq:Nnnwnw
1320 { m m m +0{ } m u{ \q_stop } }
1321 {
1322   \clist_set:Nn \l__ctex_ltj_char_range_clist {#3}
1323   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#4}
1324   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1325   \tl_if_blank:nTF {#5}
1326   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#2} }
1327   {
1328     \tl_set:Nn \l__ctex_ltj_tmp_tl {#5}
1329     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#2}
1330   }
1331   \use:x
1332   {
1333     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #1
1334     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1335     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1336     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1337   }
1338 }
1339 \clist_new:N \l__ctex_ltj_alternate_options_clist

```

(End definition for `\ctex_ltj_save_alterdate_seq:Nn` and `\ctex_ltj_save_alterdate_seq:Nnnwn`.)

`\ctex_ltj_set_alterdate_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。

```

1340 \cs_new_protected_nopar:Npn \ctex_ltj_set_alterdate_family:nn #1#2
1341   {
1342     \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1343     \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1344     \ctex_ltj_set_alterdate_family:coonn
1345     { \__ctex_ltj_alterdate_cs:n {#1} }
1346     { \l__ctex_ltj_char_range_clist }
1347     { \l__ctex_ltj_font_options_clist } {#2} {#1}
1348   }
1349 \cs_new_protected_nopar:Npn \ctex_ltj_set_alterdate_family:Nnnnn #1#2#3#4#5
1350   {
1351     \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1352     { \ctex_ltj_set_alterdate_family:nnn {#2} {#3} {#4} }
1353     \ctex_ltj_save_alterdate_family:Nnnn #1 {#2} {#3} {#4}
1354   }
1355 \cs_generate_variant:Nn \ctex_ltj_set_alterdate_family:Nnnnn { coo }

```

(End definition for `\ctex_ltj_set_alterdate_family:nn`.)

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

```

1356 \cs_new_protected_nopar:Npn \ctex_ltj_save_alterdate_family:Nnnn #1#2#3#4
1357   {
1358     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1359     \cs_gset_protected_nopar:Npx #1
1360     { \exp_not:o { #1 \ctex_ltj_set_alterdate_family:nnn {#2} {#3} {#4} } }
1361   }

```

(End definition for `\ctex_ltj_save_alterdate_family:Nnnn`.)

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

```

1362 \cs_new_protected_nopar:Npn \ctex_ltj_set_alterdate_family:nnn #1#2#3
1363   {
1364     \group_begin:
1365     \__ctex_ltj_change_encoding:
1366     \cs_set_eq:NN \CJKfamily \use_none:n
1367     \ctex_ltj_swap_cs:NN
1368     \DeclareFontShape@ \ctex_ltj_declare_alterdate_shape:nnnnn
1369     \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1370     \fontspec_set_family:Nnn \l__ctex_ltj_alterdate_family_tl {#2} {#3}
1371     \group_end:
1372   }
1373 \tl_new:N \l__ctex_ltj_alterdate_family_tl

```

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

`\ctex_ltj_swap_cs:NN` 交换两个控制序列的意义。

```

1374 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1375   {
1376     \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1377     \cs_set_eq:NN #1 #2
1378     \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1379     \cs_undefine:N \__ctex_ltj_tmp:w
1380   }

```

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

LTJFONTUID 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。

```

1381 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1382 \cs_new_protected_nopar:Npn \__ctex_ltj_update_family_uid:N #1
1383   {
1384     \int_gincr:N \g__ctex_ltj_family_int
1385     \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1386   }
1387 \int_new:N \g__ctex_ltj_family_int

```

(End definition for LTJFONTUID and __ctex_ltj_update_family_uid:N)

\ctex_ltj_declare_alter_nate_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 <series> 和第四个参数 <shape>。

```

1388 \cs_new_protected:Npn \ctex_ltj_declare_alter_nate_shape:nnnnnn #1#2#3#4#5#6
1389   {
1390     \ctex_ltj_declare_alter_nate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1391     \ctex_ltj_set_alter_nate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1392     { \l__ctex_ltj_base_family_tl } {#3} {#4}
1393     { \l_fontspec_family_tl } {#3} {#4}
1394   }

```

(End definition for \ctex_ltj_declare_alter_nate_shape:nnnnnn.)

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

```

1395 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:Nnnnnnn #1#2#3#4#5#6#7
1396   {
1397     \clist_map_inline:Nn #1
1398     {
1399       \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1400       {
1401         \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1402         \l__ctex_ltj_char_range_tl
1403       }
1404       { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1405     }
1406     \__ctex_ltj_save_alter_nate_shape:cnn
1407     { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1408     { luatexja.jfont.clear_alt_font_latex }
1409     { '\luatexluaescapestring { \CJK@encoding/#2/#3/#4 }' }
1410   }

```

(End definition for \ctex_ltj_set_alter_nate_shape:Nnnnnnn.)

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

```

1411 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1412   { m m > { \SplitArgument { \c_one } { -> } } m }
1413   { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1414 \cs_new_protected_nopar:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1415   {
1416     \ctex_ltj_set_alter_nate_shape:n
1417     {
1418       \IfNoValueTF {#4}
1419       { \int_eval:n {#3} , \int_eval:n {#3} , }
1420       {
1421         \int_eval:n { \tl_if_blank:nTF {#3} { "80 } {#3} } ,

```

```

1422         \int_eval:n { \tl_if_blank:nTF {#4} { "10FFFF } {#4} } ,
1423     }
1424     '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1425     '\luatexluaescapestring { \CJK@encoding/#1 }'
1426 }
1427 }
1428 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:n #1
1429 {
1430     \ctex_lua_now_x:n { luatexja.jfont.set_alt_font_latex ( #1 ) }
1431     \__ctex_ltj_save_alternate_shape:cnn
1432     { \__ctex_ltj_alternate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1433     { luatexja.jfont.set_alt_font_latex } {#1}
1434 }

```

(End definition for \ctex_ltj_set_alternate_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。

```

1435 \cs_new_protected_nopar:Npn \ctex_ltj_set_alternate_shape:nnN #1#2#3
1436 {
1437     \tl_map_inline:Nn #3
1438     {
1439         \ctex_ltj_set_alternate_shape:n
1440         {
1441             ##1 ,
1442             '\luatexluaescapestring { \CJK@encoding/#2 }' ,
1443             '\luatexluaescapestring { \CJK@encoding/#1 }'
1444         }
1445     }
1446 }

```

(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。

```

1447 \cs_new_protected_nopar:Npn \__ctex_ltj_save_alternate_shape:Nnn #1#2#3
1448 {
1449     \group_begin:
1450     \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1451     \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1452     \cs_set_eq:NN \luatexluaescapestring \scan_stop:
1453     \cs_gset_protected_nopar:Npx #1
1454     { \exp_not:o {#1} \exp_not:N \ctex_lua_now_x:n { #2 ( #3 ) } }
1455     \group_end:
1456 }
1457 \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。
resetalternatefont

```

1458 \keys_define:nn { ctex }
1459 {
1460     clearalternatefont .code:n =
1461     { \clist_map_function:xN {#1} \ctex_ltj_clear_alternate_font:n } ,
1462     resetalternatefont .code:n =
1463     { \clist_map_function:xN {#1} \ctex_ltj_reset_alternate_font:n } ,
1464     clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1465     resetalternatefont .default:n = \l_ctex_ltj_family_tl
1466 }
1467 \cs_new_protected_nopar:Npn \ctex_ltj_clear_alternate_font:n #1
1468 {
1469     \group_begin:
1470     \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl

```

```

1471     {
1472     \cs_if_exist_use:cT { \__ctex_ltj_alterate_cs:n { clear / #1 } }
1473     {
1474     \prop_gput:Nno \g__ctex_ltj_reset_alterate_prop
1475     {#1} { \l__ctex_ltj_base_family_tl }
1476     \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1477     \selectfont
1478     }
1479     }
1480     { \__ctex_ltj_family_unknown_warning:n {#1} }
1481 \group_end:
1482 }
1483 \cs_new_protected_nopar:Npn \ctex_ltj_reset_alterate_font:n #1
1484 {
1485 \group_begin:
1486 \prop_gpop:NnNT \g__ctex_ltj_reset_alterate_prop {#1} \CJK@family
1487 {
1488 \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1489 \use:c { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1490 \selectfont
1491 }
1492 \group_end:
1493 }
1494 \prop_new:N \g__ctex_ltj_reset_alterate_prop
1495 \cs_generate_variant:Nn \clist_map_function:nN { x }

```

(End definition for `clearalteratefont` and `resetalteratefont`. These functions are documented on page 24.)

declarecharrange 预先声明字符范围。

```

1496 \keys_define:nn { ctex }
1497 {
1498   declarecharrange .code:n = \ctex_ltj_declare_char_range:x {#1} ,
1499   declarecharrange .value_required:
1500 }
1501 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:n #1
1502 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1503 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { x }
1504 \cs_new_protected_nopar:Npn \__ctex_ltj_declare_char_range:nn #1#2
1505 { \use:x { \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } } {#2} }

```

(End definition for `declarecharrange`. This function is documented on page 24.)

```

\ctex_ltj_declare_char_range:nn #1 是名字,#2 是范围。
\g__ctex_ltj_char_range_prop
1506 \cs_new_protected_nopar:Npn \ctex_ltj_declare_char_range:nn #1#2
1507 {
1508 \tl_clear:N \l__ctex_ltj_char_range_tl
1509 \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1510 \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1511 \ctex_ltj_def_char_range_key:n {#1}
1512 \tl_clear:N \l__ctex_ltj_char_range_tl
1513 }
1514 \tl_new:N \l__ctex_ltj_char_range_tl
1515 \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。

```

1516 \NewDocumentCommand \ctex_ltj_save_char_range:n
1517 { > { \SplitArgument { \c_one } { -> } } m }
1518 { \ctex_ltj_save_char_range:nn #1 }
1519 \cs_new_protected_nopar:Npn \ctex_ltj_save_char_range:nn #1#2
1520 {
1521 \tl_put_right:Nx \l__ctex_ltj_char_range_tl

```

```

1522     { {
1523         \IfNoValueTF {#2}
1524         { \int_eval:n {#1} , \int_eval:n {#1} }
1525         {
1526             \int_eval:n { \tl_if_blank:nTF {#1} { "80 } {#1} } ,
1527             \int_eval:n { \tl_if_blank:nTF {#2} { "10FFFF } {#2} }
1528         }
1529     } }
1530 }

```

(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。

```

1531 \cs_new_protected_nopar:Npn \ctex_ltj_def_char_range_key:n #1
1532 {
1533     \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1534     {
1535         \keys_define:nn { ctex_ltj / fontspec }
1536         { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1537     }
1538 }

```

(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。

```

1539 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1540 {
1541     \tl_if_blank:nTF {#2}
1542     { \tl_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1543     {
1544         \clist_if_empty:NT \l__ctex_ltj_char_range_clist
1545         {
1546             \tl_set:Nn \l__ctex_ltj_tmp_tl { {#1} }
1547             \__ctex_ltj_char_range_parse_feature:w #2 \q_stop
1548         }
1549     }
1550 }

```

(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 `[simsun.ttc]` 设置为 `range` 的主字体,就需要使用 `range={{[simsun.ttc]}` 或者 `[][simsun.ttc]}`。下面的目的是,支持直接使用 `[simsun.ttc]` 和 `[...][simsun.ttc]`。

```

1551 \NewDocumentCommand \__ctex_ltj_char_range_parse_feature:w
1552 { +o o u { \q_stop } }
1553 {
1554     \exp_args:NNf \tl_put_right:Nn \l__ctex_ltj_tmp_tl
1555     {
1556         \IfNoValueTF {#1} { {#3} }
1557         {
1558             \IfNoValueTF {#2}
1559             { \tl_if_blank:nTF {#3} { { [ #1 ] } } { [ [ #1 ] ] {#3} } }
1560             { [ [ #1 ] ] { [ #2 ] } }
1561         }
1562     }
1563     \seq_put_right:No \l__ctex_ltj_alternate_seq { \l__ctex_ltj_tmp_tl }
1564 }

```

(End definition for `__ctex_ltj_char_range_parse_feature:w`)

15.4.3.6 其它设置

`\ctex_ltj_update_xkanjiskip:` `\l__ctex_ltj_xkanjiskip_skip` `\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 `\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。

```

1565 \cs_new_protected_nopar:Npn \ctex_ltj_update_xkanjiskip:
1566   {
1567     \skip_if_eq:nnT
1568     { \ltjgetparameter { xkanjiskip } } { \l__ctex_ltj_xkanjiskip_skip }
1569     {
1570       \skip_set:Nn \l__ctex_ltj_xkanjiskip_skip { .25 \zw plus 1pt minus 1pt }
1571       \ltjsetxkanjiskip \l__ctex_ltj_xkanjiskip_skip
1572     }
1573   }

```

注意,此时还没有实际设置字体,所以 `\zw` 是 `0pt`。

```

1574 \skip_new:N \l__ctex_ltj_xkanjiskip_skip
1575 \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, `LuaTeX-ja` 还是会使 `JAchar` 自动折行。没有看到有简单的禁用折行的办法,可能需要设置所有的 `JAchar` 的 `prebreakpenalty` 或 `postbreakpenalty` 为 `10000`:

```

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

1576 \AtBeginDocument
1577   {
1578     \ctex_appto_cmd:NnTF \verbatim@font { \CTEX@verbatim@font@hook }
1579     { } { \ctex_patch_failure:N \verbatim@font }
1580   }
1581 \cs_new_protected_nopar:Npn \CTEX@verbatim@font@hook
1582   { \ltjsetparameter { autospacing = false , autoxspacing = false } }

1583 <@@=ctex>

1584 </luatex>

```

15.4.4 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。

```

1585 <*pdfTeX||luatex>
1586 \cs_new_protected_nopar:Npn \ctex_update_default_family:
1587   {
1588     \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1589     {
1590       \group_begin:
1591       \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1592       \tl_gset:Nx \CJKfamilydefault

```



```

1593         {
1594             \str_case:onF { \familydefault }
1595             {
1596                 { \rmdefault } { \exp_not:N \CJKrmdefault }
1597                 { \sfdefault } { \exp_not:N \CJKsfdefault }
1598                 { \ttdefault } { \exp_not:N \CJKttdefault }
1599             }
1600             { \CJKfamilydefault }
1601         }
1602     \group_end:
1603 }

```

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。

```

1604 <*luatex>
1605     \ctex_ltj_ensure_default_family:
1606 </luatex>
1607 }
1608 \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。

```

1609 \tl_new:N \l__ctex_family_default_init_tl
1610 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1611 \tl_set:Nx \l__ctex_family_default_init_tl
1612 {
1613     \exp_not:N \__ctex_family_default_wrap:n
1614     { \exp_not:o { \CJKfamilydefault } }
1615 }
1616 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1617 </pdfTeX|luatex>

```

(End definition for \l__ctex_family_default_init_tl.)

15.4.5 操作系统的判断

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

```

1618 <*luatex>
1619 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1620 {
1621     \tl_gset:Nx \g__ctex_fontset_tl
1622     {
1623         \ctex_lua_now_x:n
1624         {
1625             if ~ os.name == 'windows' then ~
1626                 tex.sprint ( 'windows' )
1627             elseif ~ os.name == 'macosx' then ~
1628                 tex.sprint ( 'mac' )
1629             else ~
1630                 tex.sprint ( 'fandol' )
1631             end
1632         }
1633     }
1634 }
1635 </luatex>

```

pdf \TeX 和 X \TeX 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!

```

1636 <*xetex|pdfTeX>

```

```

1637 \cs_new_protected_nopar:Npn \ctex_detected_platform:
1638 {
1639   \file_if_exist:nTF { /dev/null }
1640   {
1641     \file_if_exist:nTF { nul: }
1642     {
1643       \file_if_exist:nTF { \c__ctex_engine_file_tl }
1644       { \ctex_if_macosx:TF { mac } { windows } }
1645       { \ctex_if_macosx:TF { mac } { fandol } }
1646     }
1647     { \ctex_if_macosx:TF { mac } { fandol } }
1648   }
1649   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
1650 }
1651 \tex_uppercase:D \exp_after:wN
1652 {
1653   \exp_after:wN \tl_const:Nn \exp_after:wN \c__ctex_engine_file_tl
1654   \exp_after:wN { \g_file_current_name_tl }
1655 }

```

(End definition for \ctex_detected_platform:.)

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

```

1656 \cs_new_protected_nopar:Npn \ctex_if_macosx:TF #1#2
1657 {
1658   \file_if_exist:nTF { \c__ctex_macosx_file_tl }
1659   { \tl_gset:Nn \g__ctex_fontset_tl {#1} }
1660   { \tl_gset:Nn \g__ctex_fontset_tl {#2} }
1661 }
1662 \tl_const:Nn \c__ctex_macosx_file_tl { /mach_kernel }
1663 </xetex|pdftex>

```

(End definition for \ctex_if_macosx:TF.)

15.4.6 hyperref 兼容性处理

现在处理各个引擎下的 PDF 中文书签问题。根据编译引擎与文件编码的不同, ctex 向 hyperref 传递适当的参数, 完成中文书签的正确设置。用户仍需要自己载入 hyperref 宏包。

\ctex_hypersetup:n 如果已经载入 hyperref 宏包, 则直接使用其定义设置选项; 否则 \ctex_hypersetup:n 的效果与 \PassOptionsToPackage 一致, 只传递宏包参数。如果用户不载入 hyperref 宏包, 相关参数即被丢弃。

```

1664 <*classstyle>
1665 \@ifpackageloaded { hyperref }
1666 {
1667   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1668   { \hypersetup {#1} }
1669 }
1670 {
1671   \cs_new_protected_nopar:Npn \ctex_hypersetup:n #1
1672   { \PassOptionsToPackage {#1} { hyperref } }
1673 }
1674 \ctex_hypersetup:n { colorlinks = true }
1675 </classstyle>

```

(End definition for \ctex_hypersetup:n.)

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。

```

1676 <*pdftex>
1677 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1678 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
1679 {
1680   \ctex_hypersetup:n { CJKbookmarks = true }
1681   \ctex_if_pdfmode:TF
1682   { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1683   {
1684     \ctex_at_end_package:nn { hyperref }
1685     {
1686       \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1687       {
1688         \AtBeginShipoutFirst
1689         { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1690       }
1691       { \RequirePackage { xCJK2uni } }
1692     }
1693   }
1694 }
1695 { \ctex_hypersetup:n { pdfencoding = unicode } }
1696 </pdftex>

```

在 X_ƎT_ƎX 下, hyperref 在处理带有非 ASCII 字符和 \% 的书签时有问题¹⁵。事实上, hyperref 在驱动文件 h_xet_ex.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}{%
}{%
\EdefUnescapeString\HyPsd@temp#1%
\ifxetex
\let\HyPsd@UnescapedString\HyPsd@temp
\StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
{utf16be}{ascii-print}{%
\EdefEscapeString\HyPsd@temp\HyPsd@temp
\global\let#1\HyPsd@temp
\HyPsd@EscapeTeX#1%
\Hy@unicodefalse
}{%
\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 X_DVIPDFMX 的警告¹⁶:

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

X_ƎT_ƎX 的维护者 Khaled Hosny 已经注意到了这个问题¹⁷。需要注意的是, h_xet_ex.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。如果 hyperref 先于 C_TƎX 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 X_ƎT_ƎX 下更保险的 pdfencoding 选项来设置。为了与 X_ƎT_ƎX 下的行为一致(使用 \HyPsd@LoadUnicode 载入 puenc.def), 在 Lua_TƎX 下也启用这个选项。

```

1697 <*xetex|luatex>
1698 \ctex_hypersetup:n { pdfencoding = unicode }
1699 </xetex|luatex>

1700 <*pdftex|xetex|luatex>

```

¹⁵<https://github.com/CTeX-org/ctex-kit/issues/39>

¹⁶<http://project.ktug.org/dvipdfmx/mailman/dvipdfmx/2009-December/000153.html>

¹⁷<http://tug.org/pipermail/tex-live/2013-December/034613.html>

15.4.7 CJKfntef、xeCJKfntef 相关设置

CT_EX 套件对 pdfT_EX 与 X_ET_EX 引擎,分别载入 CJKfntef 或 xeCJKfntef 宏包,并关闭宏包默认的彩色等多余格式。

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。

```

1701 <pdfTeX>\RequirePackage { CJKfntef } \normalem
1702 <xetex>\RequirePackage { xeCJKfntef }
1703 <*luatex>
1704 \msg_new:nnn { ctex } { fntef-not-available }
1705 { Functions~ of~ `CJKfntef'~ is~ not~ available~ in~ LuaLaTeX. }
1706 \msg_warning:nn { ctex } { fntef-not-available }
1707 </luatex>
1708 \clist_map_inline:nn
1709 { underdot , underline , underdblline , underwave , sout , xout }
1710 <*pdfTeX|xetex>
1711 {
1712 <pdfTeX> \tl_clear:c { CJK#1color }
1713 <xetex> \keys_set:nn { xeCJK / options } { #1 / format = { } }
1714 \cs_new_protected_nopar:cpx { CTEX#1 }
1715 {
1716 \msg_warning:nnnn { ctex } { deprecated-command } { \exp_not:c { CTEX#1 } }
1717 { You~ can~ use~ the~ command~ with~ prefix~ \exp_not:N \CJK~ instead. }
1718 \exp_not:c { CJK#1 }
1719 }
1720 }
1721 \cs_new_protected_nopar:Npn { \CTEXfilltwosides }
1722 {
1723 \msg_warning:nnnn { ctex } { deprecated-environment } { CTEXfilltwosides }
1724 { You~ can~ use~ `CJKfilltwosides'~ environment~ instead. }
1725 \CJKfilltwosides
1726 }
1727 \cs_new_protected_nopar:Npn { \endCTEXfilltwosides } { \endCJKfilltwosides }
1728 </pdfTeX|xetex>
1729 <*luatex>
1730 { \cs_new_eq:cN { CTEX#1 } \use:n }
1731 \cs_new_eq:NN \CTEXfilltwosides \use_none:n
1732 \cs_new_eq:NN \endCTEXfilltwosides \prg_do_nothing:
1733 </luatex>
1734 <*pdfTeX>
1735 \clist_map_inline:nn
1736 {
1737 underdotbasesep , underdotsep , underlinebasesep ,
1738 underlinesep , underdbllinesep , underdbllinebasesep ,
1739 underwavebasesep , underwavesep , southeight ,
1740 underdotcolor , underwavecolor , underlinecolor ,
1741 underdbllinecolor , soutcolor , xoutcolor
1742 }
1743 {
1744 \cs_new_eq:cc { CTEX#1 } { CJK#1 }
1745 \cs_set_nopar:cpx { CJK#1 } { \exp_not:c { CTEX#1 } }
1746 }
1747 </pdfTeX>

```

15.4.8 \ccwd 的更新

\ctex_update_ccwd:

```

\ccwd
1748 \cs_new_protected_nopar:Npn \ctex_update_ccwd:
1749 <*pdfTeX|xetex>
1750 {
1751 \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1752 \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }

```

```

1753 }
1754 </pdfTeX|xetex>
1755 <*luatex>
1756 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1757 </luatex>
1758 \dim_new:N \ccwd

```

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

\ctex_update_ccglue: 更新字间距。

```

1759 \cs_new_protected_nopar:Npn \ctex_update_ccglue:
1760 <*pdfTeX>
1761 {
1762   \cs_set_protected_nopar:Npn \CJKglue
1763   { \skip_horizontal:N \l__ctex_ccglue_skip }
1764 }
1765 </pdfTeX>
1766 <*xetex>
1767 { \xeCJKsetup { CJKglue = { \skip_horizontal:N \l__ctex_ccglue_skip } } }
1768 </xetex>
1769 <*luatex>
1770 { \ltjsetkanjiskip \l__ctex_ccglue_skip }
1771 </luatex>
1772 \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

```

1773 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1774 {
1775 <*pdfTeX|xetex>
1776   \if_meaning:w \CJKglue \l__ctex_ccglue:
1777   \prg_return_false: \else: \prg_return_true: \fi:
1778 </pdfTeX|xetex>
1779 <*luatex>
1780   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1781   { \prg_return_false: } { \prg_return_true: }
1782 </luatex>
1783 }

```

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 @@。

```

1784 <*pdfTeX|xetex>
1785 \ctex_at_end:n { \cs_new_eq:NN \l__ctex_ccglue: \CJKglue }
1786 </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 XeTeX 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 XeTeX 可以用 \fontcharwd 来改进。而在 pdfTeX 下, 若使用 zhmetrics 技术, 所有的汉字共享同一个 .tfm, \fontcharwd 也就没有意义。在 LuaTeX 下, LuaTeX-ja 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 \zw 作为汉字宽度。

```

1787 \cs_new_protected_nopar:Npn \ctex_update_em_unit:
1788 <pdfTeX|xetex> { \dim_set:Nn \ccwd { \f@size \p@ } }
1789 <luatex> { \dim_set:Nn \ccwd { \zw } }

```

(End definition for \ctex_update_em_unit:.)

15.4.9 其它

`\ctex_add_to_selectfont:n` `\CTEX@selectfont@hook` `\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 `LuaTeX-ja` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。 `everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。

```

1790 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1791 {
1792   \cs_set_protected_nopar:Npx \CTEX@selectfont@hook
1793   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1794 }
1795 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1796 \cs_new_eq:Nc \__ctex_save_selectfont: { selectfont ~ }
1797 \use:n
1798 {
1799   \ExplSyntaxOff
1800   \ctex_preto_cmd:NnTF \selectfont { \CTEX@selectfont@hook }
1801   {
1802     \tl_put_left:Nn \@EverySelectfont@Init
1803     { \cs_set_eq:cN { selectfont ~ } \__ctex_save_selectfont: }
1804   }
1805   { \ctex_patch_failure:N \selectfont }
1806 }
1807 \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。

```

1808 <*pdfTeX>
1809 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1810 </pdfTeX>
1811 <*xetex|luatex>
1812 \EverySelectfont { \CTEX@selectfont@hook }
1813 </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。

```

1814 <*luatex>
1815 \ctex_add_to_selectfont:n
1816 {
1817   \ctex_ltj_select_font:
1818   \ctex_ltj_select_alternate_font:
1819   \ctex_ltj_update_xkanjiskip:
1820 }
1821 \tl_set:Nn \CJK@family { song } \selectfont
1822 \tl_clear:N \CJK@family
1823 </luatex>

```

space 在导言区或正文中设置忽略空格方式。 `pdfTeX` 和 `XYTeX` 下初始设置为 `auto`, `LuaTeX` 下是无效选项。

```

1824 \keys_define:nn { ctex }
1825 {
1826 <*pdfTeX|xetex>
1827   space .choice: ,
1828   space / true .code:n =
1829 <pdfTeX> { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
1830 <xetex> { \xeCJKsetup { CJKspace = true } } ,
1831   space / auto .code:n =
1832 <pdfTeX> { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
1833 <xetex> { \xeCJKsetup { CJKspace = false } } ,
1834   space / false .code:n =
1835 <pdfTeX> { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,

```

```

1836 <xetex>      { \xeCJKsetup { CJKspace = false } } ,
1837     space .default:n = { true } ,
1838     space .initial:n = { auto }
1839 </pdfptex|xetex>
1840 <*luatex>
1841     space .code:n =
1842     { \msg_warning:nn { ctex } { invalid-option } }
1843 </luatex>
1844 }

```

(End definition for space. This function is documented on page 27.)

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-ja 设置的是字体的默认 JFM, 只会影响到之后设置的字体。

```

1845 \keys_define:nn { ctex }
1846 {
1847     punct .code:n =
1848     {
1849         \tl_set:Nx \l__ctex_punct_tl { #1 }
1850 <pdfptex>      \punctstyle { \l__ctex_punct_tl }
1851 <xetex>        \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
1852 <luatex>      \ctex_mono_jfm:o { \l__ctex_punct_tl }
1853     } ,
1854     punct .default:n = { quanjiao } ,
1855 }

```

(End definition for punct. This function is documented on page 27.)

```

1856 </pdfptex|xetex|luatex>
1857 <*class|style>

```

15.4.10 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

1858 \pdfptex_if_engine:TF
1859 {
1860     \tl_set:Nx \l__ctex_encoding_tl { \l__ctex_encoding_tl }
1861     \ctex_file_input:n { ctex-engine-pdfptex.def }
1862 }
1863 {
1864     \tl_set:Nn \l__ctex_encoding_tl { UTF8 }
1865     \xetex_if_engine:TF
1866     { \ctex_file_input:n { ctex-engine-xetex.def } }
1867     { \ctex_file_input:n { ctex-engine-luatex.def } }
1868 }

```

15.5 字距与缩进

autoindent autoindent 也是可以用在正文中的选项, 意义与宏包选项 option/autoindent 相同。

```

1869 \keys_define:nn { ctex }
1870 {
1871     autoindent .choice: ,
1872     autoindent .default:n = { true } ,
1873     autoindent / true .code:n =
1874     {
1875         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
1876         \ctex_select_size:
1877     } ,
1878     autoindent / false .code:n =
1879     { \tl_clear:N \l__ctex_autoindent_tl } ,
1880     autoindent / unknown .code:n =

```

```

1881     {
1882       \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
1883       \ctex_select_size:
1884     }
1885   }

```

(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。

```

1886 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
1887 \cs_new_protected_nopar:Npn \ctex_select_size:
1888   { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

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

\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。

```

1889 \cs_new_protected_nopar:Npn \ctex_update_size:
1890   {
1891     \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
1892     {
1893       \ctex_update_stretch:
1894       \ctex_update_parindent:
1895     }
1896     { \ctex_update_ziju: }
1897   }
1898 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
1899 \tl_new:N \l__ctex_ziju_tl
1900 \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。

```

1901 \ctex_add_to_selectfont:n
1902   { \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`。

```

1903 \keys_define:nn { ctex }
1904   {
1905     linestretch .code:n =
1906       {
1907         \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
1908         \ctex_select_size:
1909       } ,
1910     linestretch .value_required:
1911   }
1912 \tl_new:N \l__ctex_line_stretch_tl
1913 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

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

\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。


```

1914 \cs_new_protected_nopar:Npn \ctex_update_stretch:
1915 {
1916   \ctex_update_em_unit:
1917   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
1918   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
1919   {
1920     \ctex_if_ccglue_touched:TF
1921     { \ctex_update_ccwd: }
1922     {
1923       \dim_set:Nn \l__ctex_tmp_dim
1924       { \baselinestretch \etex_glueexpr:D \f@baselineskip \scan_stop: }
1925       \skip_set:Nn \l__ctex_ccglue_skip
1926       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
1927       \ctex_update_ccglue:
1928     }
1929   }
1930   {
1931     \int_set:Nn \l__ctex_tmp_int
1932     { \etex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
1933     \tex_divide:D \l__ctex_tmp_int \ccwd
1934     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero
1935     {
1936       \skip_set:Nn \l__ctex_ccglue_skip
1937       {
1938         \c_zero_dim plus \dim_eval:n
1939         {
1940           ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
1941           \l__ctex_tmp_int
1942         }
1943       }
1944     }
1945     { \skip_zero:N \l__ctex_ccglue_skip }
1946     \ctex_update_ccglue:
1947   }
1948 }

```

(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。

```

1949 \cs_new_protected_nopar:Npn \ctex_update_parindent:
1950 {
1951   \tl_if_empty:NF \l__ctex_autoindent_tl
1952   {
1953     \dim_compare:nNnF \parindent = \c_zero_dim
1954     { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
1955   }
1956 }

```

(End definition for \ctex_update_parindent:.)

\ziju 若参数为 0, 则恢复正常间距。

```

1957 \NewDocumentCommand \ziju { m }
1958 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
1959 \cs_new_protected_nopar:Npn \ctex_ziju:n #1
1960 {
1961   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
1962   \ctex_select_size:
1963 }

```

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

\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。

```

1964 \cs_new_protected_nopar:Npn \ctex_update_ziju:
1965 {
1966   \ctex_update_em_unit:
1967   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
1968   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
1969   \dim_compare:nNnTF \ccwd > \c_zero_dim

```

伸展量保证行内的剩余空白能够被均匀地填充到汉字之间,收缩的最大限度是让当前行还能够再挤下一个汉字并且不会出现负间距。由 \TeX 决定伸展还是收缩。

```

1970   {
1971     \dim_set:Nn \l__ctex_tmp_dim
1972     { \linewidth - \ccwd + \l__ctex_ziju_dim }
1973     \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
1974     \tex_divide:D \l__ctex_tmp_int \ccwd
1975     \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。

```

1976     \dim_compare:nNnF \parindent = \c_zero_dim
1977     {
1978       \int_compare:nNnF \l__ctex_tmp_int < \c_three
1979       { \int_sub:Nn \l__ctex_tmp_int { \c_two } }
1980     }
1981     \skip_set:Nn \l__ctex_ccglue_skip
1982     {
1983       \l__ctex_ziju_dim
1984       plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
1985       minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
1986       { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + \c_one ) }
1987     }
1988   }
1989   { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
1990   \ctex_update_ccglue:

```

字距设置得比较大时,为了尽量保证段首缩进能够与下一行对齐,应该需要相应地加上或者减去伸缩值。但是这里并不清楚 \TeX 是伸展还是收缩,之前以“当前行是否还放得下一个汉字”为标准加上或减去伸缩值的做法也未必与实际结果一致,所以只好还是设置为 $2\ccwd$ 。

```

1991   \ctex_update_parindent:
1992 }
1993 \dim_new:N \l__ctex_ziju_dim

```

(End definition for `\ctex_update_ziju:`)

`\CTEXindent`
`\CTEXnoindent` 过时命令。

```

1994 \NewDocumentCommand \CTEXindent { }
1995 {
1996   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXindent }
1997   { \parindent is~ set~ to~ 2\ccwd. }
1998   \ctex_update_ccwd: \dim_set:Nn \parindent { 2 \ccwd }
1999 }
2000 \NewDocumentCommand \CTEXnoindent { }
2001 {
2002   \msg_warning:nnnn { ctex } { deprecated-command } { \CTEXnoindent }
2003   { \parindent is~ set~ to~ 0pt. }
2004   \dim_zero:N \parindent
2005 }

```

(End definition for `\CTEXindent` and `\CTEXnoindent`. These functions are documented on page 27.)

15.6 中文数字与日期

```
2006 \PassOptionsToPackage { encoding = \l__ctex_encoding_tl } { zhnumber }
2007 \RequirePackage { zhnumber }
```

\chinese

```
2008 \cs_new_eq:NN \chinese \zhnum
2009 \cs_new_eq:NN \Chinese \chinese
2010 \cs_new_eq:NN \CTEXcounter \use_none:n
```

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

\CTEXnumber**\CTEXdigits**

```
2011 \NewDocumentCommand \CTEXnumber { m m }
2012 { \protected@edef #1 { \zhnumber {#2} } }
2013 \NewDocumentCommand \CTEXdigits { m m }
2014 { \protected@edef #1 { \zhdigits {#2} } }
```

(End definition for \CTEXnumber and \CTEXdigits. These functions are documented on page 23.)

today

```
2015 \cs_set_eq:NN \CTEX@todayold \today
2016 \keys_define:nn { ctex }
2017 {
2018   today .choice: ,
2019   today / old .code:n =
2020     { \cs_set_eq:NN \today \CTEX@todayold } ,
2021   today / small .code:n =
2022     {
2023       \cs_set_eq:NN \today \zhtoday
2024       \zhnumsetup { time = Arabic }
2025     } ,
2026   today / big .code:n =
2027     {
2028       \cs_set_eq:NN \today \zhtoday
2029       \zhnumsetup { time = Chinese }
2030     } ,
2031   today / unknown .code:n =
2032     { \msg_error:nxx { ctex } { today-undef } {#1} }
2033 }
2034 \msg_new:nnnn { ctex } { today-undef }
2035 { Today~format~`#1'~is~undefined. }
2036 { Available~today~formats~are~`old',~`small',~and~`big'. }
```

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

15.7 其它中文标题定义

```
2037 \keys_define:nn { ctex }
2038 {
2039   contentsname .tl_set:N = \contentsname ,
2040   listfigurename .tl_set:N = \listfigurename ,
2041   listtablename .tl_set:N = \listtablename ,
2042   figurename .tl_set:N = \figurename ,
2043   tablename .tl_set:N = \tablename ,
2044   abstractname .tl_set:N = \abstractname ,
2045   indexname .tl_set:N = \indexname ,
2046   appendixname .tl_set:N = \appendixname ,
2047   <article> bibname .tl_set:N = \refname
2048   <book|report> bibname .tl_set:N = \bibname
2049 }
2050 <*style>
2051 \msg_new:nnn { ctex } { ctexbibname }
2052 {
```

```

2053     Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.\\
2054     The~key~`bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2055   }
2056   \tl_if_exist:NTF \bibname
2057   { \keys_define:nn { ctex } { bibname .tl_set:N = \bibname } }
2058   {
2059     \tl_if_exist:NTF \refname
2060     { \keys_define:nn { ctex } { bibname .tl_set:N = \refname } }
2061     {
2062       \msg_warning:nn { ctex } { ctexbibname }
2063       \keys_define:nn { ctex } { bibname .tl_set:N = \ctexbibname }
2064     }
2065   }
2066 </style>
2067 </class|style>

```

15.8 中文化的标题结构

本节内容在 CTEX 文档类或打开 heading 选项下生效。

```
2068 <*class|heading>
```

15.8.1 定义标题格式选项

```
\c__ctex_headings_seq
```

```

2069 \seq_new:N \c__ctex_headings_seq
2070 \seq_gset_from_clist:Nn \c__ctex_headings_seq
2071 {
2072 <article> part , section , subsection , subsubsection ,
2073 <book|report> part , chapter , section , subsection , subsubsection ,
2074 paragraph , subparagraph
2075 }

```

(End definition for \c__ctex_headings_seq.)

```
\__ctex_initial_heading:n
```

```

2076 \cs_new_protected_nopar:Npn \__ctex_initial_heading:n #1
2077 {
2078   \tl_new:c { CTEX@pre#1 }
2079   \tl_new:c { CTEX@post#1 }
2080   \tl_const:cx { CTEX@the#1 }
2081   {
2082     \exp_not:c { CTEX@pre#1 }
2083     \exp_not:c { CTEX@the#1 }
2084     \exp_not:c { CTEX@post#1 }
2085   }
2086   \tl_const:cx { CTEX@#1name }
2087   {
2088     \exp_not:c { CTEX@#1nameformat }
2089     \exp_not:c { CTEX@pre#1 }
2090     \exp_not:N \tl_if_empty:NTF \exp_not:c { CTEX@#1numberformat }
2091     { \exp_not:c { CTEX@the#1 } }
2092     {
2093       \group_begin:
2094         \exp_not:c { CTEX@#1numberformat }
2095         \exp_not:c { CTEX@the#1 }
2096       \group_end:
2097     }
2098     \exp_not:c { CTEX@post#1 }
2099     \exp_not:c { CTEX@#1aftername }
2100   }
2101 }

```

(End definition for `_ctex_initial_heading:n`)

`_ctex_def_heading_keys:n`

```

2102 \cs_new_protected_nopar:Npn \_ctex_def_heading_keys:n #1
2103 {
2104   \tl_put_right:Nx \l__ctex_tmp_tl
2105   {
2106     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2107     #1 / name .code:n =
2108     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2109     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2110     #1 / format .tl_set:N = \exp_not:c { CTEX@#1@format } ,
2111     #1 / nameformat .tl_set:N = \exp_not:c { CTEX@#1@nameformat } ,
2112     #1 / numberformat .tl_set:N = \exp_not:c { CTEX@#1@numberformat } ,
2113     #1 / aftername .tl_set:N = \exp_not:c { CTEX@#1@aftername } ,
2114     #1 / titleformat .tl_set:N = \exp_not:c { CTEX@#1@titleformat } ,
2115     #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,
2116     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2117     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2118     #1 / format+ .code:n =
2119     { \tl_put_right:Nn \exp_not:c { CTEX@#1@format } { #####1 } } ,
2120     #1 / nameformat+ .code:n =
2121     { \tl_put_right:Nn \exp_not:c { CTEX@#1@nameformat } { #####1 } } ,
2122     #1 / numberformat+ .code:n =
2123     { \tl_put_right:Nn \exp_not:c { CTEX@#1@numberformat } { #####1 } } ,
2124     #1 / aftername+ .code:n =
2125     { \tl_put_right:Nn \exp_not:c { CTEX@#1@aftername } { #####1 } } ,
2126     #1 / titleformat+ .code:n =
2127     { \tl_put_right:Nn \exp_not:c { CTEX@#1@titleformat } { #####1 } } ,
2128     #1 / beforekip .initial:n = \c_zero_skip ,
2129     #1 / afterskip .initial:n = \c_zero_skip ,
2130     #1 / indent .initial:n = \c_zero_dim ,
2131     #1 / beforekip .value_required: ,
2132     #1 / afterskip .value_required: ,
2133     #1 / indent .value_required: ,
2134   }
2135 }

```

(End definition for `_ctex_def_heading_keys:n`)

`ctex_assign_heading_name:nn`
`_ctex_assign_heading_name:nnn`

`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。

```

2136 \NewDocumentCommand \ctex_assign_heading_name:nn
2137 { m > { \SplitArgument { \c_one } { , } } +m }
2138 { \_ctex_assign_heading_name:nnn {#1} #2 }
2139 \cs_new_protected:Npn \_ctex_assign_heading_name:nnn #1#2#3
2140 {
2141   \tl_set:cn { CTEX@pre#1 } {#2}
2142   \IfNoValueTF {#3}
2143   { \tl_clear:c { CTEX@post#1 } }
2144   { \tl_set:cn { CTEX@post#1 } {#3} }
2145 }

```

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

```

2146 \tl_clear:N \l__ctex_tmp_tl
2147 \seq_map_inline:Nn \c__ctex_headings_seq
2148 {
2149   \_ctex_initial_heading:n {#1}
2150   \_ctex_def_heading_keys:n {#1}
2151 }
2152 \use:x { \keys_define:nn { ctex } { \exp_not:o { \l__ctex_tmp_tl } } }

```

```
2153 \tl_clear:N \l__ctex_tmp_tl
```

pagestyle 只在 ctexbook 和 ctexrep 下有定义。

```
2154 (*book|report)
2155 \keys_define:nn { ctex }
2156 {
2157   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2158   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle
2159 }
2160 </book|report>
```

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

15.8.2 标准标题命令的修改

15.8.2.1 part 的标题

```
2161 <@@=>
```

\part

```
2162 (*article)
2163 \renewcommand\part{%
2164   \if@noskipsec \leavevmode \fi
2165   \par
2166   % \addvspace{4ex}%
2167   \@tempskipa \CTEX@part@beforeskip \relax
2168   \ifdim \@tempskipa <\z@
2169     \@tempskipa -\@tempskipa \@afterindentfalse
2170   \else
2171     \@afterindenttrue
2172   \fi
2173   \addvspace{\@tempskipa}%
2174   \secdef\@part\@spart}
2175 </article>
```

(End definition for \part.)

\@part

```
2176 (*article)
2177 \def\@part[#1]#2{%
2178   \ifnum \c@secnumdepth >\m@ne
2179     \refstepcounter{part}%
2180     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2181     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2182   \else
2183     \addcontentsline{toc}{part}{#1}%
2184   \fi
2185   {\interlinepenalty \@M
2186   % \normalfont \parindent \z@ \raggedright
2187   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2188   \ifnum \c@secnumdepth >\m@ne
2189     % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2190     \CTEX@partname
2191   \fi
2192   % \huge\bfseries #2%
2193   \CTEX@part@titleformat{#2}%
2194   \markboth{}{\par}%
2195   \nobreak
2196   % \vskip 3ex
2197   \vskip \CTEX@part@afterskip
2198   \@afterheading}
2199 </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。

```

2200 <*book|report>
2201 \def\@part[#1]#2{%
2202   \thispagestyle{\CTEX@part@pagestyle}%
2203   \ifnum \c@secnumdepth >-2\relax
2204     \refstepcounter{part}%
2205     % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2206     \addcontentsline{toc}{part}{\CTEXthepart\hspace{1em}#1}%
2207   \else
2208     \addcontentsline{toc}{part}{#1}%
2209   \fi
2210   \markboth{}{}%
2211   {\interlinepenalty \@M
2212   % \normalfont \centering
2213   \normalfont \CTEX@part@format
2214   \ifnum \c@secnumdepth >-2\relax
2215     % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2216     \CTEX@partname
2217     \fi
2218     % \Huge\bfseries #2\par}%
2219     \CTEX@part@titleformat{#2}\par}%
2220   \@endpart}
2221 </book|report>

```

(End definition for `\@part`.)

`\@spart`

```

2222 <*article>
2223 \def\@spart#1{%
2224   {\interlinepenalty \@M
2225   % \normalfont \parindent \z@ \raggedright
2226   \normalfont \parindent \CTEX@part@indent \CTEX@part@format
2227   % \huge \bfseries #1\par}%
2228   \CTEX@part@titleformat{#1}\par}%
2229   \nobreak
2230   % \vskip 3ex
2231   \vskip \CTEX@part@afterskip
2232   \@afterheading}
2233 </article>
2234 <*book|report>
2235 \def\@spart#1{%
2236   {\interlinepenalty \@M
2237   % \normalfont \centering
2238   \normalfont \CTEX@part@format
2239   % \Huge \bfseries #1\par}%
2240   \CTEX@part@titleformat{#1}\par}%
2241   \@endpart}
2242 </book|report>

```

(End definition for `\@spart`.)

15.8.2.2 chapter 的标题

```

2243 <*book|report>

```

`\@chapter`

```

2244 \def\@chapter[#1]#2{%
2245   \ifnum \c@secnumdepth >\m@ne
2246   <book> \ifmainmatter
2247     \refstepcounter{chapter}%
2248     % \typeout{\@chapapp\space\thechapter.}%
2249     \typeout{\CTEXthechapter}%

```

```

2250     \addcontentsline{toc}{chapter}
2251 %     {\protect\numberline{\thechapter}#1}%
2252     {\protect\numberline{\CTEXthechapter\hspace{0.3em}}#1}%
2253 <book>     \else
2254 <book>     \addcontentsline{toc}{chapter}{#1}%
2255 <book>     \fi
2256     \else
2257     \addcontentsline{toc}{chapter}{#1}%
2258     \fi
2259     \chaptermark{#1}%
2260     \addtocontents{lof}{\protect\addvspace{10\p@}}%
2261     \addtocontents{lot}{\protect\addvspace{10\p@}}%
2262     \if@twocolumn
2263     \topnewpage[\@makechapterhead{#2}]%
2264     \else
2265     \makechapterhead{#2}%
2266     \@afterheading
2267     \fi}

```

(End definition for \@chapter.)

\@makechapterhead

```

2268 \def\@makechapterhead#1{%
2269   \thispagestyle{\CTEX@chapter@pagestyle}%
2270   % \vspace*{50\p@}%
2271   \@tempskipa \CTEX@chapter@beforeskip \relax
2272   \ifdim \@tempskipa <\z@
2273     \@tempskipa -\@tempskipa \@afterindentfalse
2274   \else
2275     \@afterindenttrue
2276   \fi
2277   \vspace*{\@tempskipa}%
2278   % {\normalfont \parindent \z@ \raggedright
2279   {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2280     \ifnum \c@secnumdepth >\m@ne
2281 <book>     \if@mainmatter
2282 %     \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2283     \CTEX@chaptername
2284 <book>     \fi
2285     \fi
2286     \interlinepenalty\@M
2287 % \Huge \bfseries #1\par\nobreak
2288     \CTEX@chapter@titleformat{#1}\par\nobreak
2289 % \vskip 40\p@
2290     \vskip \CTEX@chapter@afterskip
2291   }}

```

(End definition for \@makechapterhead.)

\@makeschapterhead

```

2292 \def\@makeschapterhead#1{%
2293   \thispagestyle{\CTEX@chapter@pagestyle}%
2294   % \vspace*{50\p@}%
2295   \@tempskipa \CTEX@chapter@beforeskip \relax
2296   \ifdim \@tempskipa <\z@
2297     \@tempskipa -\@tempskipa \@afterindentfalse
2298   \else
2299     \@afterindenttrue
2300   \fi
2301   \vspace*{\@tempskipa}%
2302   % {\normalfont \parindent \z@ \raggedright
2303   {\normalfont \parindent \CTEX@chapter@indent \CTEX@chapter@format
2304     \interlinepenalty\@M
2305 % \Huge \bfseries #1\par\nobreak
2306     \CTEX@chapter@titleformat{#1}\par\nobreak

```



```

2307 % \vskip 40\p@
2308 \vskip \CTEX@chapter@afterskip
2309 }

```

(End definition for \@makeschapterhead.)

```
2310 </book|report>
```

15.8.2.3 section 类的标题

\@secntformat

```

2311 \def\@secntformat#1{%
2312 \ifundefined{CTEX@#1name}%
2313 {\csname the#1\endcsname\quad}%
2314 {\csname CTEX@#1name\endcsname}}

```

(End definition for \@secntformat.)

\@sect

```

2315 \def\@sect#1#2#3#4#5#6[#7]#8{%
2316 \ifnum #2>\c@secnumdepth
2317 \let\@svsec\@empty
2318 \else
2319 \refstepcounter{#1}%
2320 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2321 \fi
2322 \@tempskipa #5\relax
2323 \ifdim \@tempskipa>\z@
2324 \begingroup
2325 #6{%
2326 \@hangfrom{\hskip #3\relax\@svsec}%
2327 % \interlinepenalty \@M #8\@par}%
2328 \interlinepenalty \@M
2329 \csname CTEX@#1@titleformat\endcsname{#8}\@par}%
2330 \endgroup
2331 \csname #1mark\endcsname{#7}%
2332 \addcontentsline{toc}{#1}{%
2333 \ifnum #2>\c@secnumdepth \else
2334 % \protect\numberline{\csname the#1\endcsname}%
2335 \protect\numberline{\ifundefined{CTEXthe#1}%
2336 \csname the#1\endcsname}%
2337 \csname CTEXthe#1\endcsname}}%
2338 \fi
2339 #7}%
2340 \else
2341 \def\@svsechd{%
2342 #6{\hskip #3\relax
2343 % \@svsec #8}%
2344 \@svsec \csname CTEX@#1@titleformat\endcsname{#8}}%
2345 \csname #1mark\endcsname{#7}%
2346 \addcontentsline{toc}{#1}{%
2347 \ifnum #2>\c@secnumdepth \else
2348 % \protect\numberline{\csname the#1\endcsname}%
2349 \protect\numberline{\ifundefined{CTEXthe#1}%
2350 \csname the#1\endcsname}%
2351 \csname CTEXthe#1\endcsname}}%
2352 \fi
2353 #7}}%
2354 \fi
2355 \@xsect{#5}}

```

(End definition for \@sect.)

`\@ssect` `\@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`。

```

2356 \def\@ssect#1#2#3#4#5{%
2357   \@tempskipa #3\relax
2358   \ifdim \@tempskipa>\z@
2359     \begingroup
2360       #4{%
2361         \@hangfrom{\hskip #1}%
2362         % \interlinepenalty \@M #5\@par}%
2363         \interlinepenalty \@M
2364         \CTEX@titleformat@n{#5}\@par}%
2365       \endgroup
2366   \else
2367   % \def\@svsechd{#4{\hskip #1\relax #5}}%
2368   \def\@svsechd{#4{\hskip #1\relax \CTEX@titleformat@n{#5}}}%
2369   \fi
2370   \@xsect{#3}}

```

(End definition for \@ssect.)

```
2371 <@@=ctex>
```

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

```

2372 \cs_new_protected_nopar:Npn \CTEX@set@titleformat@n #1
2373 { \cs_set_eq:NC \CTEX@titleformat@n { CTEX@#1@titleformat } }
2374 \cs_new_eq:NN \CTEX@titleformat@n \use:n

```

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

```

2375 \int_zero:N \l__ctex_tmp_int
2376 \clist_map_inline:nn
2377 { section , subsection , subsubsection , paragraph , subparagraph }
2378 {
2379   \int_incr:N \l__ctex_tmp_int
2380   \cs_gset_protected_nopar:cpx {#1}
2381   {
2382     \exp_not:N \@startsection {#1}
2383     { \int_use:N \l__ctex_tmp_int }
2384     { \exp_not:c { CTEX@#1@indent } }
2385     { \exp_not:c { CTEX@#1@beforeskip } }
2386     { \exp_not:c { CTEX@#1@afterskip } }
2387     {
2388       \CTEX@set@titleformat@n {#1}
2389       \exp_not:N \normalfont \exp_not:c { CTEX@#1@format }
2390     }
2391   }
2392 }

```

15.8.2.4 附录标题

```

2393 \tl_new:N \CTEX@preappendix
2394 \tl_new:N \CTEX@postappendix
2395 \keys_define:nn { ctex }
2396 { appendix .meta:nn = { ctex / appendix } {#1} }
2397 \keys_define:nn { ctex / appendix }
2398 {
2399   name .code:n = { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2400   number .tl_set:N = \CTEX@appendixnumber
2401 }

```

`\appendix`

```

2402 \cs_new_eq:NN \CTEX@save@appendix \appendix
2403 \cs_gset_protected_nopar:Npn \appendix
2404 {
2405   \CTEX@save@appendix
2406   <*article>
2407     \gdef \CTEX@presection { \CTEX@preappendix }
2408     \gdef \CTEX@thesection { \CTEX@appendixnumber }
2409     \gdef \CTEX@postsection { \CTEX@postappendix }
2410   </article>
2411   <*book|report>
2412     \gdef \CTEX@prechapter { \CTEX@preappendix }
2413     \gdef \CTEX@thechapter { \CTEX@appendixnumber }
2414     \gdef \CTEX@postchapter { \CTEX@postappendix }
2415   </book|report>
2416 }

```

(End definition for \appendix.)

15.8.3 目录标签的宽度

\numberline

```

2417 \cs_new_protected:Npn \CTEX@toc@width@n #1
2418 {
2419   \hbox_set:Nn \l__ctex_tmp_box {#1}
2420   \dim_set:Nn \@tempdima
2421   {
2422     \dim_max:nn { \@tempdima }
2423     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / \c_two }
2424   }
2425 }
2426 \group_begin:
2427 \char_set_catcode_other:N \#
2428 \use:n
2429 {
2430   \group_end:
2431   \ExplSyntaxOff
2432   \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2433   { \ctex_patch_failure:N \numberline }
2434   \ifpackageloaded { tocloft }
2435   { }
2436   {
2437     \ctex_at_end_package:nn { tocloft }
2438     {
2439       \ctex_preto_cmd:NnTF \numberline { \CTEX@toc@width@n {#1} } { }
2440       { \ctex_patch_failure:N \numberline }
2441     }
2442   }
2443   \ExplSyntaxOn
2444 }

```

(End definition for \numberline.)

15.8.4 页眉信息的修改

\ps@headings

```

2445 <*article>
2446 \ctex_patch_cmd:Nnn \ps@headings { \thesection } { \CTEXthesection }
2447 \if@twoside
2448 \ctex_patch_cmd:Nnn \ps@headings { \thesubsection } { \CTEXthesubsection }
2449 \fi:
2450 </article>
2451 <*book|report>

```

```

2452 \ctex_patch_cmd:Nnn \ps@headings
2453 { \@chapapp\ \thechapter.~\ } { \CTEXthechapter \quad }
2454 \if@twoside
2455 \ctex_patch_cmd:Nnn \ps@headings { \thesection.~\ } { \CTEXthesection \quad }
2456 \fi:
2457 </book|report>

```

(End definition for \ps@headings.)

`\ps@fancy` 这里对 `fancyhdr` 宏包打补丁。原来 `fancyhdr` 宏包中使用 `\thesection` 等宏表示页眉中的章节编号,这里改用 `ctex` 包所用的 `\CTEXthesection` 系列宏。

```

2458 \ctex_at_end_package:nn { fancyhdr }
2459 {
2460 <*article>
2461 \ctex_patch_cmd:Nnn \ps@fancy
2462 { \thesection \hskip 1em \relax } { \CTEXthesection \quad }
2463 \ctex_patch_cmd:Nnn \ps@fancy
2464 { \thesubsection \hskip 1em \relax } { \CTEXthesubsection \quad }
2465 </article>
2466 <*book|report>
2467 \ctex_patch_cmd:Nnn \ps@fancy
2468 <book> { \@chapapp\ \thechapter.~\ } { \if@mainmatter \CTEXthechapter \quad \fi }
2469 <report> { \@chapapp\ \thechapter.~\ } { \CTEXthechapter \quad }
2470 \ctex_patch_cmd:Nnn \ps@fancy
2471 { \thesection.~\ } { \CTEXthesection \quad }
2472 </book|report>
2473 }

```

(End definition for \ps@fancy.)

15.8.5 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。

```

2474 \cs_new_protected_nopar:Npn \CTEX@setcurrentlabel@n #1
2475 {
2476 \protected@edef \@currentlabel
2477 {
2478 \cs_if_exist:CTF { CTEX@the#1 }
2479 { \exp_args:cc { p@#1 } { CTEX@the#1 } }
2480 { \exp_not:o { \@currentlabel } }
2481 }
2482 }

```

(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。

```

2483 \cs_new_protected_nopar:Npn \ctex_varioref_hook:
2484 {
2485 \seq_map_inline:Nn \c__ctex_headings_seq
2486 { \ctex_fix_varioref_label:n { ##1 } }
2487 }

```

(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>`。`LATEX` 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。

```
2488 \cs_new_protected_nopar:Npn \ctex_fix_varioref_label:n #1
2489 {
2490   \tl_if_empty:cT { p@#1 }
2491   { \exp_args:Nno \labelformat {#1} { \cs:w CTEX@the#1 \cs_end: } }
2492 }
```

(End definition for _ctex_fix_varioref_label:n)

如果 `varioref` 已经被载入,则使用它来设置。

```
2493 \@ifpackageloaded { varioref }
2494 { \ctex_varioref_hook: }
2495 {
2496   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
2497   \RenewDocumentCommand \refstepcounter { m }
2498   {
2499     \CTEX@save@refstepcounter {#1}
2500     \CTEX@setcurrentlabel@n {#1}
2501   }
2502   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
2503 }
```

15.8.6 载入 $\langle scheme \rangle$ 文件

```
2504 \ctex_scheme_input:o { \l_{}_ctex_scheme_tl }
2505 \</class|heading>
```

15.8.7 标题格式的 `scheme` 定义

下面使用 $\text{CT}_{\text{E}}\text{X}$ 文档类的设置方式, `plain` 模拟标准文档类直接定义或以 `\@startsection` 设定的章节标题格式, `chinese` 汉化的标题格式。

```
2506 \<*scheme&(article|book|report)>
2507 \keys_set:nn { ctex / part }
2508 {
2509 \<*plain>
2510   name      = \partname \space ,
2511   number    = \thepart ,
2512 \</plain>
2513 \<*article>
2514   nameformat = \Large \bfseries ,
2515   afterskip  = 3ex ,
2516 \<*plain>
2517   format     = \raggedright ,
2518   aftername  = \par \nobreak ,
2519   titleformat = \huge \bfseries ,
2520   beforeskip = -4ex
2521 \</plain>
2522 \<*chinese>
2523   number     = \chinese { part } ,
2524   format     = \centering ,
2525   aftername  = \quad ,
2526   titleformat = \Large \bfseries ,
2527   beforeskip = 4ex
2528 \</chinese>
2529 \</article>
2530 \<*book|report>
2531   format     = \centering ,
2532   nameformat = \huge \bfseries ,
2533   aftername  = \par \vskip 20 \p@ ,
2534   pagestyle  = plain ,
```

```

2535 <*plain>
2536     titleformat = \Huge \bfseries
2537 </plain>
2538 <*chinese>
2539     titleformat = \huge \bfseries
2540 </chinese>
2541 </book|report>
2542 }

2543 <*book|report>
2544 \keys_set:nn { ctex / chapter }
2545 {
2546     pagestyle     = plain ,
2547     nameformat    = \huge \bfseries ,
2548     afterskip     = 40 \p@ ,
2549 <*plain>
2550     name          = \chaptername \space ,
2551     number        = \thechapter ,
2552     format        = \raggedright ,
2553     aftername     = \par \nobreak \vskip 20 \p@ ,
2554     titleformat   = \Huge \bfseries ,
2555     beforekip     = -50 \p@
2556 </plain>
2557 <*chinese>
2558     number        = \chinese { chapter } ,
2559     format         = \centering ,
2560     aftername     = \quad ,
2561     titleformat   = \huge \bfseries ,
2562     beforekip     = 50 \p@
2563 </chinese>
2564 }
2565 </book|report>

2566 \keys_set:nn { ctex / section }
2567 {
2568     number        = \thesection ,
2569     aftername     = \quad ,
2570     afterskip     = 2.3ex \@plus .2ex ,
2571 <*plain>
2572     format        = \Large \bfseries ,
2573     beforekip     = -3.5ex \@plus -1ex \@minus -.2ex
2574 </plain>
2575 <*chinese>
2576     format        = \Large \bfseries \centering ,
2577     beforekip     = 3.5ex \@plus 1ex \@minus .2ex
2578 </chinese>
2579 }

2580 \keys_set:nn { ctex / subsection }
2581 {
2582     number        = \thesubsection ,
2583     format        = \large \bfseries ,
2584     aftername     = \quad ,
2585     afterskip     = 1.5ex \@plus .2ex ,
2586 <*plain>
2587     beforekip     = -3.25ex \@plus -1ex \@minus -.2ex
2588 </plain>
2589 <*chinese>
2590     beforekip     = 3.25ex \@plus 1ex \@minus .2ex
2591 </chinese>
2592 }

2593 \keys_set:nn { ctex / subsubsection }
2594 {
2595     number        = \thesubsubsection ,
2596     format        = \normalsize \bfseries ,
2597     aftername     = \quad ,
2598     afterskip     = 1.5ex \@plus .2ex ,

```

```

2599 <*plain>
2600     beforekip = -3.25ex \@plus -1ex \@minus -.2ex
2601 </plain>
2602 <*chinese>
2603     beforekip = 3.25ex \@plus 1ex \@minus .2ex
2604 </chinese>
2605 }

```

```

2606 \keys_set:nn { ctex / paragraph }
2607 {
2608     number      = \theparagraph ,
2609     format      = \normalsize \bfseries ,
2610     aftername   = \quad
2611 }

```

```

2612 \keys_set:nn { ctex / subparagraph }
2613 {
2614     number      = \thesubparagraph ,
2615     format      = \normalsize \bfseries ,
2616     aftername   = \quad
2617 }

```

处理 sub3section 与 sub4section 的格式。

```

2618 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2619 {
2620     \keys_set:nn { ctex / paragraph }
2621     {
2622 <*plain>
2623         beforekip = -3.25ex \@plus -1ex \@minus -.2ex ,
2624 </plain>
2625 <*chinese>
2626         beforekip = 3.25ex \@plus 1ex \@minus .2ex ,
2627 </chinese>
2628         afterskip = 1ex \@plus .2ex
2629     }
2630 }
2631 {
2632     \keys_set:nn { ctex / paragraph }
2633     {
2634         beforekip = 3.25ex \@plus 1ex \@minus .2ex ,
2635         afterskip = -1em
2636     }
2637 }
2638 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_three
2639 {
2640     \keys_set:nn { ctex / subparagraph }
2641     {
2642 <*plain>
2643         beforekip = -3.25ex \@plus -1ex \@minus -.2ex ,
2644 </plain>
2645 <*chinese>
2646         beforekip = 3.25ex \@plus 1ex \@minus .2ex ,
2647 </chinese>
2648         afterskip = 1ex \@plus .2ex
2649     }
2650 }
2651 {
2652     \keys_set:nn { ctex / subparagraph }
2653     {
2654         beforekip = 3.25ex \@plus 1ex \@minus .2ex ,
2655         afterskip = -1em
2656     }
2657 }
2658 \int_compare:nNnTF \g__ctex_section_depth_flag > \c_two
2659 { \keys_set:nn { ctex / subparagraph } { indent = \c_zero_dim } }
2660 { \keys_set:nn { ctex / subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

2661 \keys_set:nn { ctex / appendix }
2662 ⟨*article⟩
2663   { number      = \@Alph \c@section }
2664 ⟨/article⟩
2665 ⟨*book|report⟩
2666   {
2667     name        = \appendixname \space ,
2668     number      = \@Alph \c@chapter
2669   }
2670 ⟨/book|report⟩
2671 ⟨/scheme&(article|book|report)⟩

```

15.8.8 ctex.sty 的 heading 选项

```

2672 ⟨*style⟩
2673 \msg_new:nnn { ctex } { not-standard-class }
2674 {
2675   None~of~the~standard~document~classes~was~loaded.\\
2676   Heading~`#1'~is~selected.\\
2677   ctex~may~not~work~as~expected.
2678 }
2679 \bool_if:NTF \l__ctex_heading_bool
2680 {
2681   \clist_map_inline:nn { article , book , report }
2682   {
2683     \@ifclassloaded {#1}
2684     { \clist_map_break:n { \tl_const:Nn \c__ctex_class_tl {#1} } } { }
2685   }
2686   \tl_if_exist:NF \c__ctex_class_tl
2687   {
2688     \cs_if_exist:NTF \chapter
2689     {
2690       \cs_if_exist:NF \if@mainmatter
2691       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
2692       \tl_const:Nn \c__ctex_class_tl { book }
2693     }
2694     { \tl_const:Nn \c__ctex_class_tl { article } }
2695     \msg_warning:nx { ctex } { not-standard-class } { \c__ctex_class_tl }
2696   }
2697   \ctex_file_input:n { ctex- \c__ctex_class_tl .def }
2698 }
2699 { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
2700 ⟨/style⟩

```

15.8.9 标题配置文件

```

2701 ⟨*UTF8|GBK⟩
2702 \keys_set:nn { ctex }
2703 {
2704   contentsname = 目录 ,
2705   listfigurename = 插图 ,
2706   listtablename = 表格 ,
2707   figurename = 图 ,
2708   tablename = 表 ,
2709   abstractname = 摘要 ,
2710   indexname = 索引 ,
2711   bibname = 参考文献 ,
2712   appendixname = 附录
2713 }
2714 \keys_if_exist:nnT { ctex / part } { name }
2715 {
2716   \keys_set:nn { ctex / part } { name = { 第 , 部分 } }

```



```

2717 \keys_if_exist:nnT { ctex / chapter } { name }
2718 { \keys_set:nn { ctex / chapter } { name = { 第 , 章 } } }
2719 }
2720 </UTF8|GBK>

```

15.9 chinese 方案的其他设置

```
2721 <*scheme&chinese>
```

chinese 在标准文档类下的页面格式总采用 headings。

```
2722 <article|book|report>\pagestyle { headings }
```

汉化标题名字。

```

2723 \str_if_eq:onTF { \l__ctex_encoding_tl } { GBK }
2724 { \ctex_file_input:n { ctex-name-gbk.cfg } }
2725 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

日期格式。

```
2726 \keys_set:nn { ctex } { today = small }
```

若用户未设置宏包选项 zihao, 则设置 \normalsize 为五号字。

```

2727 \int_compare:nNnF \g__ctex_font_size_flag > \c_minus_one
2728 { \cs_gset_eq:NN \g__ctex_font_size_flag \c_zero }

```

若用户未设置宏包选项 linespread, 则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。

```

2729 \fp_compare:nNnT { \l__ctex_line_spread_fp } ? { \c_zero_fp }
2730 { \keys_set:nn { ctex / option } { linespread = 1.3 } }

```

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

```

2731 \ctex_if_autoindent_touched:F
2732 { \keys_set:nn { ctex } { autoindent = true } }

```

若 ctex 宏包与标准文档类及其衍生文档类联用, 则将载入 indentfirst 宏包, 实现章节标题后首个段落的段首缩进。

```

2733 <*(article|book|report)>
2734 \clist_map_inline:nn { article , book , report }
2735 {
2736   \ifclassloaded {#1}
2737   { \RequirePackage{ indentfirst } } { }
2738 }
2739 </!(article|book|report)>
2740 </scheme&chinese>

```

15.10 中文字号

```
2741 <*class|style|ctexsize>
```

\zihao

```

2742 \NewDocumentCommand \zihao { m }
2743 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

```

(End definition for \zihao. This function is documented on page 21.)

\ctex_zihao:n

```

2744 \cs_new_protected_nopar:Npn \ctex_zihao:n #1
2745 {
2746   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
2747   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
2748   { \msg_error:nnn { ctex } { fontsize } {#1} }
2749 }
2750 \msg_new:nnnn { ctex } { fontsize }

```

```

2751 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
2752 {
2753   The~old~font~size~is~used~if~you~continue.\\
2754   The~available~font~sizes~are~listed~as~follow.\\
2755   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
2756 }

```

(End definition for \ctex_zihao:n.)

15.10.1 定义中文字号

\c__ctex_font_size_prop 基础行距是字号的 1.2 倍,采用 ϵ -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确¹⁸。

```

2757 \prop_new:N \c__ctex_font_size_prop
2758 \seq_new:N \c__ctex_font_size_seq
2759 \cs_new_protected_nopar:Npn \__ctex_save_font_size:nn #1#2
2760 {
2761   \use:x
2762   {
2763     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
2764     {
2765       { \dim_to_decimal:n {#2} }
2766       { \dim_to_decimal:n { (#2) * \c_six / \c_five } }
2767     }
2768   }
2769   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
2770 }
2771 \clist_map_inline:nn
2772 {
2773   { 8 } { 5 bp } ,
2774   { 7 } { 5.5 bp } ,
2775   { -6 } { 6.5 bp } ,
2776   { 6 } { 7.5 bp } ,
2777   { -5 } { 9 bp } ,
2778   { 5 } { 10.5 bp } ,
2779   { -4 } { 12 bp } ,
2780   { 4 } { 14 bp } ,
2781   { -3 } { 15 bp } ,
2782   { 3 } { 16 bp } ,
2783   { -2 } { 18 bp } ,
2784   { 2 } { 22 bp } ,
2785   { -1 } { 24 bp } ,
2786   { 1 } { 26 bp } ,
2787   { -0 } { 36 bp } ,
2788   { 0 } { 42 bp }
2789 }
2790 { \__ctex_save_font_size:nn #1 }

```

(End definition for \c__ctex_font_size_prop.)

\ctex_declare_math_sizes:mnnn

```

2791 \cs_new_protected_nopar:Npn \ctex_declare_math_sizes:mnnn #1#2#3#4
2792 {
2793   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
2794   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
2795 }

```

(End definition for \ctex_declare_math_sizes:mnnn.)

__ctex_get_font_sizes:Nn

¹⁸<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

2796 \cs_new_protected_nopar:Npn \__ctex_get_font_sizes:Nn #1#2
2797 {
2798   \tl_clear:N #1
2799   \tl_map_inline:nn {#2}
2800   {
2801     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
2802     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
2803     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
2804   }
2805 }

```

(End definition for __ctex_get_font_sizes:Nn.)

```

2806 \clist_map_inline:nn
2807 {
2808   { 8 }{ 8 }{ 5pt }{ 5pt } ,
2809   { 7 }{ 7 }{ 5pt }{ 5pt } ,
2810   { -6 }{ -6 }{ 5pt }{ 5pt } ,
2811   { 6 }{ 6 }{ 5pt }{ 5pt } ,
2812   { -5 }{ -5 }{ 6pt }{ 5pt } ,
2813   { 5 }{ 5 }{ 7pt }{ 5pt } ,
2814   { -4 }{ -4 }{ 8pt }{ 6pt } ,
2815   { 4 }{ 4 }{ 5 }{ 6 } ,
2816   { -3 }{ -3 }{ -4 }{ -5 } ,
2817   { 3 }{ 3 }{ 4 }{ 5 } ,
2818   { -2 }{ -2 }{ -3 }{ -4 } ,
2819   { 2 }{ 2 }{ 3 }{ 4 } ,
2820   { -1 }{ -1 }{ -2 }{ -3 } ,
2821   { 1 }{ 1 }{ 2 }{ 3 } ,
2822   { -0 }{ -0 }{ -1 }{ -2 } ,
2823   { 0 }{ 0 }{ 1 }{ 2 }
2824 }
2825 { \ctex_declare_math_sizes:nmmn #1 }

```

15.10.2 修改默认字号大小

\ctex_set_font_size:Nnn

```

2826 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
2827 {
2828   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
2829   { \exp_after:wN \__ctex_set_font_size:nmNn \l__ctex_font_size_tl #1 {#3} }
2830   { \msg_error:nmm { ctex } { fontsize } {#2} }
2831 }
2832 \cs_new_protected:Npn \__ctex_set_font_size:nmNn #1#2#3#4
2833 { \cs_set_protected_nopar:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

```

(End definition for \ctex_set_font_size:Nnn.)

```

2834 \if_case:w \g__ctex_font_size_flag
2835 \ctex_file_input:n { ctex-c5size.clo }
2836 \or:
2837 \ctex_file_input:n { ctex-cs4size.clo }
2838 \fi:
2839 </class|style|ctexsize>
2840 <*c5size>
2841 \ctex_set_font_size:Nnn \normalsize { 5 }
2842 {
2843   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2844   \abovedisplayshortskip \z@ \@plus3\p@
2845   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2846   \belowdisplayskip \abovedisplayskip
2847   \let\@listi\@listI
2848 }
2849 \ctex_set_font_size:Nnn \small { -5 }

```

```

2850 {
2851   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
2852   \abovedisplayshortskip \z@ \@plus2\p@
2853   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
2854   \def\@listi{\leftmargin\leftmargini
2855             \topsep 4\p@ \@plus2\p@ \@minus2\p@
2856             \parsep 2\p@ \@plus\p@ \@minus\p@
2857             \itemsep \parsep}
2858   \belowdisplayskip \abovedisplayskip
2859 }
2860 \ctex_set_font_size:Nnn \footnotesize { 6 }
2861 {
2862   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
2863   \abovedisplayshortskip \z@ \@plus\p@
2864   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
2865   \def\@listi{\leftmargin\leftmargini
2866             \topsep 3\p@ \@plus\p@ \@minus\p@
2867             \parsep 2\p@ \@plus\p@ \@minus\p@
2868             \itemsep \parsep}
2869   \belowdisplayskip \abovedisplayskip
2870 }
2871 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
2872 \ctex_set_font_size:Nnn \tiny { 7 } { }
2873 \ctex_set_font_size:Nnn \large { -4 } { }
2874 \ctex_set_font_size:Nnn \Large { -3 } { }
2875 \ctex_set_font_size:Nnn \LARGE { -2 } { }
2876 \ctex_set_font_size:Nnn \huge { 2 } { }
2877 \ctex_set_font_size:Nnn \Huge { 1 } { }
2878 </c5size>
2879 <*cs4size>
2880 \ctex_set_font_size:Nnn \normalsize { -4 }
2881 {
2882   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
2883   \abovedisplayshortskip \z@ \@plus3\p@
2884   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2885   \belowdisplayskip \abovedisplayskip
2886   \let\@listi\@listI
2887 }
2888 \ctex_set_font_size:Nnn \small { 5 }
2889 {
2890   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
2891   \abovedisplayshortskip \z@ \@plus3\p@
2892   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
2893   \def\@listi{\leftmargin\leftmargini
2894             \topsep 9\p@ \@plus3\p@ \@minus5\p@
2895             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
2896             \itemsep \parsep}
2897   \belowdisplayskip \abovedisplayskip
2898 }
2899 \ctex_set_font_size:Nnn \footnotesize { -5 }
2900 {
2901   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
2902   \abovedisplayshortskip \z@ \@plus3\p@
2903   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
2904   \def\@listi{\leftmargin\leftmargini
2905             \topsep 6\p@ \@plus2\p@ \@minus2\p@
2906             \parsep 3\p@ \@plus2\p@ \@minus\p@
2907             \itemsep \parsep}
2908   \belowdisplayskip \abovedisplayskip
2909 }
2910 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
2911 \ctex_set_font_size:Nnn \tiny { -6 } { }
2912 \ctex_set_font_size:Nnn \large { -3 } { }
2913 \ctex_set_font_size:Nnn \Large { -2 } { }
2914 \ctex_set_font_size:Nnn \LARGE { 2 } { }
2915 \ctex_set_font_size:Nnn \huge { -1 } { }
2916 \ctex_set_font_size:Nnn \Huge { 1 } { }

```

```

2917 </cs4size>
2918 <ctexsize>\normalsize
2919 <*class|style>

```

15.11 更新行距

`\l__ctex_line_spread_fp` 被设置了才有必要更新行距和 `\footnotesep`。

```

2920 \fp_compare:nNnF { \l__ctex_line_spread_fp } ? { \c_zero_fp }
2921 {
2922   \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距，可能导致脚注的间距与行距不协调，需要调整 `\footnotesep`。标准文档类对 `\footnotesep` 的设置是，字体大小为 `\footnotesize` 时 `\strutbox` 的高度（默认值是 `.7\baselineskip`）。我们沿用这个设置方法，只需要更新具体的大小。

```

2923   \group_begin: \footnotesize \exp_args:NNNo \group_end:
2924   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
2925 }

```

(End definition for \footnotesep.)

激活默认字体大小，更新行距、`\parindent` 和 `\CJKglue`。

```

2926 \normalsize

```

15.12 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。

```

2927 \NewDocumentCommand \CTeX { }
2928 { \ifincsname CTeX \else: C \TeX \fi: }
2929 \AtBeginDocument
2930 {
2931   \cs_if_exist_use:NT \pdfstringdefDisableCommands
2932   { { \tl_set:Nn \CTeX { CTeX } } }
2933 }

```

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

`captiondelimitter` 过时选项。

```

2934 \keys_define:nn { ctex }
2935 {
2936   captiondelimitter .code:n =
2937   {
2938     \msg_warning:nnn { ctex } { deprecated-option }
2939     { You~can~load~the~package~`caption'~to~get~its~functionality. }
2940   }
2941 }

```

(End definition for captiondelimitter.)

```

2942 </class|style>

```

15.12.1 列表环境的缩进

`\verse` 只在使用文档类的时候修改诗歌和引用环境的缩进。

```

\quotation
2943 <*scheme&!generic&chinese>
2944 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
2945 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
2946 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
2947 </scheme&!generic&chinese>

(End definition for \verse and \quotation.)

2948 <*class|style>

```

15.12.2 其他兼容性修改

`\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。

```

2949 \group_begin:
2950 \char_set_catcode_other:N \#
2951 \use:n
2952 {
2953   \group_end:
2954   \ExplSyntaxOff
2955   \ctex_at_end_package:nn { breqn }
2956   {
2957     \ctex_patch_cmd:NnnTF \end { \csname end#1 \endcsname }
2958     {
2959       \csuse { @end@#1@hook }
2960       \csname end#1 \endcsname
2961     } { }
2962     {
2963       \ctex_preto_cmd:NnTF \end { \csuse { @end@#1@hook } }
2964       { } { \ctex_patch_failure:N \end }
2965     }
2966   }
2967   \ExplSyntaxOn
2968 }

```

(End definition for `\end`.)

15.13 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。

```

2969 \cs_new_protected_nopar:Npn \ctex_fontset_error:n #1
2970 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
2971 \msg_new:nnn { ctex } { fontset-unavailable }
2972 { 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。

```

2973 \cs_new_protected_nopar:Npn \ctex_load_fontset:
2974 {
2975   \tl_if_empty:NTF \g__ctex_fontset_tl
2976   {
2977     \ctex_detected_platform:
2978     \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2979   }
2980   {
2981     \file_if_exist:nTF { ctex-fontset- \g__ctex_fontset_tl .def }
2982     { \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def } }
2983     {
2984       \use:x
2985       {
2986         \ctex_detected_platform:
2987         \msg_error:nxxx { ctex } { fontset-not-found }
2988         { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
2989       }
2990       \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
2991     }
2992   }
2993 }
2994 \@onlypreamble \ctex_load_fontset:
2995 \msg_new:nnnn { ctex } { fontset-not-found }
2996 {
2997   CTeX~fontset~`#1'~could~not~be~found.\\
2998   Fontset~`#2'~will~be~used~instead.
2999 }
3000 { 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。

```

3001 \keys_define:nn { ctex }
3002 {
3003   fontset .code:n =
3004   {
3005     \ctex_if_preamble:TF
3006     {
3007       \str_if_eq_x:nnTF {#1} { none }
3008       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
3009       {
3010         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
3011         {
3012           \tl_gset:Nx \g__ctex_fontset_tl {#1}
3013           \ctex_load_fontset:
3014         }
3015         {
3016           \msg_error:nxxx { ctex } { fontset-loaded }
3017           { \g__ctex_fontset_tl } {#1}
3018         }
3019       }
3020     }
3021     { \msg_error:nn { ctex } { fontset-only-preamble } }
3022   }
3023 }
3024 \msg_new:nnnn { ctex } { fontset-loaded }
3025 {
3026   CTeX~fontset~`#1'~has~been~loaded.
3027   \str_if_eq:nnF {#1} {#2} { \\ Fontset~`#2'~will~be~ignored. }
3028 }
3029 { Only~one~fontset~can~be~loaded~in~the~preamble. }
3030 \msg_new:nnn { ctex } { fontset-only-preamble }
3031 {
3032   The~`fontset'~option~can~be~used~only~in~preamble.
3033 }

```

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

载入中文字库。

```
3034 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
3035 { \ctex_load_fontset: }
```

15.14 宏包配置文件

15.14.1 ctex.cfg

```
3036 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
3037 </classstyle>
3038 <*config>
3039 %%
3040 </config>
```

15.14.2 ctexopts.cfg

这里仅为配置文件示例:使用 Windows Vista 或以后版本的字体设置。

```
3041 <*ctexopts>
3042 %%
3043 %% \keys_set:nn { ctex / option } { fontset = windowsnew }
3044 </ctexopts>
```

15.15 字体定义文件

15.15.1 传统定义方式

```
3045 <*c19|c70>
3046 %%
3047 %% Chinese characters
3048 %%
3049 <c19>%% character set: GBK (extension of GB 2312)
3050 <c70>%% character set: Unicode
3051 %% font encoding: Unicode
3052 %%
3053 </c19|c70>

3054 <rm&c19>\DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
3055 <rm&c70>\DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
3056 <sf&c19>\DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
3057 <sf&c70>\DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
3058 <tt&c19>\DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
3059 <tt&c70>\DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

3060 <*rm>
3061 <*c19>
3062 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
3063 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
3064 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
3065 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
3066 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3067 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
3068 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
3069 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
3070 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
3071 </c19>
3072 <*c70>
3073 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
3074 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
3075 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
3076 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
3077 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
```



```

3078 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
3079 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
3080 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
3081 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
3082 </c70>
3083 </rm>
3084 < *sf>
3085 < *c19>
3086 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
3087 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
3088 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
3089 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
3090 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3091 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
3092 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
3093 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
3094 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
3095 </c19>
3096 < *c70>
3097 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
3098 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
3099 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
3100 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
3101 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3102 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
3103 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
3104 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
3105 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
3106 </c70>
3107 </sf>
3108 < *tt>
3109 < *c19>
3110 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
3111 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
3112 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
3113 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
3114 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3115 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
3116 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
3117 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
3118 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
3119 </c19>
3120 < *c70>
3121 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
3122 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
3123 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
3124 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
3125 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
3126 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
3127 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
3128 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
3129 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
3130 </c70>
3131 </tt>
3132 < *fontset>

```

15.15.2 ctex-fontset-windows.def, ctex-fontset-windowsnew.def, ctex-fontset-windowsold.def

ctex 包利用 C:\boot.ini 文件的存在性来判断是否使用 Windows XP 及以前的版本，分别载入新旧字体设置。

```

3133 < *windows>
3134 \file_if_exist:nTF { C:/boot.ini }
3135 { \ctex_file_input:n { ctex-fontset-windowsold.def } }
3136 { \ctex_file_input:n { ctex-fontset-windowsnew.def } }
3137 </windows>

```

旧的 Windows 字体设置使用黑体作为无衬线体，楷体和仿宋是 GB2312 编码；新的 Windows 字体设置使用微软雅黑作为无衬线体，楷体和仿宋是大字库。

```

3138 <*windowsnew,windowssold)
3139 \pdfTeX_if_engine:TF
3140 {
3141     \ctex_zhmap_case:nnn
3142     {
3143         \ctex_punct_set:n { windows }
3144         \setCJKmainfont
3145         [ BoldFont = simhei.ttf , ItalicFont = simkai.ttf ] { simsun.ttc }
3146 <*windowssold)
3147         \setCJKsansfont { simhei.ttf }
3148         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3149 </windowssold)

```

Windows 8 以后，微软雅黑由原来的 .ttf 后缀改为 .ttc 后缀，需要加以区分。

```

3150 <*windowsnew)
3151     \file_if_exist:nTF { C:/Windows/Fonts/msyh.ttc }
3152     {
3153         \setCJKsansfont [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3154         \setCJKfamilyfont { zhyahei }
3155         [ BoldFont = msyhbd.ttc ] { msyh.ttc }
3156     }
3157     {
3158         \setCJKsansfont [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3159         \setCJKfamilyfont { zhyahei }
3160         [ BoldFont = msyhbd.ttf ] { msyh.ttf }
3161     }
3162     \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
3163     \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhyahei } { zhyaheib }
3164 </windowsnew)
3165     \setCJKmonofont { simfang.ttf }
3166     \setCJKfamilyfont { zhkai } { simkai.ttf }
3167     \setCJKfamilyfont { zhfs } { simfang.ttf }
3168     \setCJKfamilyfont { zhsong } { simsun.ttc }
3169     \setCJKfamilyfont { zhhei } { simhei.ttf }
3170     \setCJKfamilyfont { zhli } { simli.ttf }
3171     \setCJKfamilyfont { zhyou } { simyou.ttf }
3172     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3173     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3174     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3175     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3176 }
3177 {
3178     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
3179     \ctex_punct_set:n { windows }
3180     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3181     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3182     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3183 }
3184 {
3185     \tl_set:Nn \CJKrmdefault { rm }
3186     \tl_set:Nn \CJKsfdefault { sf }
3187     \tl_set:Nn \CJKttdefault { tt }
3188 }
3189 }
3190 {
3191 <*windowssold)
3192     \setCJKmainfont
3193     [ BoldFont = SimHei , ItalicFont = KaiTi_GB2312 ] { SimSun }
3194     \setCJKsansfont { SimHei }
3195     \setCJKmonofont { FangSong_GB2312 }
3196     \setCJKfamilyfont { zhkai } { KaiTi_GB2312 }
3197     \setCJKfamilyfont { zhfs } { FangSong_GB2312 }
3198 </windowssold)
3199 <*windowsnew)

```

```

3200 \setCJKmainfont
3201 [ BoldFont = SimHei , ItalicFont = KaiTi ] { SimSun }
3202 \setCJKsansfont
3203 [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3204 \setCJKmonofont { FangSong }
3205 \setCJKfamilyfont { zhkai } { KaiTi }
3206 \setCJKfamilyfont { zhfs } { FangSong }
3207 </windowsnew>
3208 \setCJKfamilyfont { zhsong } { SimSun }
3209 \setCJKfamilyfont { zhhei } { SimHei }
3210 \setCJKfamilyfont { zhli } { LiSu }
3211 \setCJKfamilyfont { zhyou } { YouYuan }
3212 <*windowsnew>
3213 \setCJKfamilyfont { zhyahei }
3214 [ BoldFont = { *~Bold } ] { Microsoft~YaHei }
3215 </windowsnew>
3216 }
3217 </windowsnew,windowssold>

```

15.15.3 ctex-fontset-adobe.def

```
3218 <*adobe>
```

\pdfmapline 不支持 OpenType 字体，因而 adobe 字体集在 pdf 模式下就没有定义。fandol 的情况类似。

```

3219 \pdfif_engine:TF
3220 {
3221   \ctex_if_pdfmode:TF
3222   { \ctex_fontset_error:n { adobe } }
3223   {
3224     \ctex_zhmap_case:nnn
3225     {
3226       \setCJKmainfont
3227       [
3228         cmap = UniGB-UTF16-H ,
3229         BoldFont = AdobeHeitiStd-Regular.otf ,
3230         ItalicFont = AdobeKaitiStd-Regular.otf
3231       ] { AdobeSongStd-Light.otf }
3232       \setCJKsansfont [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3233       \setCJKmonofont [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3234       \setCJKfamilyfont { zhsong }
3235       [ cmap = UniGB-UTF16-H ] { AdobeSongStd-Light.otf }
3236       \setCJKfamilyfont { zhhei }
3237       [ cmap = UniGB-UTF16-H ] { AdobeHeitiStd-Regular.otf }
3238       \setCJKfamilyfont { zhkai }
3239       [ cmap = UniGB-UTF16-H ] { AdobeKaitiStd-Regular.otf }
3240       \setCJKfamilyfont { zhfs }
3241       [ cmap = UniGB-UTF16-H ] { AdobeFangsongStd-Regular.otf }
3242       \ctex_punct_set:n { adobe }
3243       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3244       \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3245       \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3246       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3247       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3248     }
3249     {
3250       \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
3251       \ctex_punct_set:n { adobe }
3252       \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3253       \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3254       \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3255     }
3256     { \ctex_fontset_error:n { adobe } }
3257   }
3258 }
3259 {

```

```

3260 \setCJKmainfont
3261 [
3262     BoldFont = AdobeHeitiStd-Regular ,
3263     ItalicFont = AdobeKaitiStd-Regular
3264 ] { AdobeSongStd-Light }
3265 \setCJKsansfont { AdobeHeitiStd-Regular}
3266 \setCJKmonofont { AdobeFangsongStd-Regular}
3267 \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
3268 \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
3269 \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
3270 \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
3271 }
3272 </adobe>
3273 <*fandol>

```

15.15.4 ctex-fontset-fandol.def

```

3274 \pdfTeX_if_engine:TF
3275 {
3276     \ctex_if_pdfmode:TF
3277     { \ctex_fontset_error:n { fandol } }
3278     {
3279         \ctex_zhmap_case:nnn
3280         {
3281             \setCJKmainfont
3282             [
3283                 cmap = UniGB-UTF16-H ,
3284                 BoldFont = FandolSong-Bold.otf ,
3285                 ItalicFont = FandolKai-Regular.otf
3286             ] { FandolSong-Regular.otf }
3287             \setCJKsansfont
3288             [
3289                 cmap = UniGB-UTF16-H ,
3290                 BoldFont = FandolHei-Bold.otf
3291             ] { FandolHei-Regular.otf }
3292             \setCJKmonofont [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3293             \setCJKfamilyfont { zhsong }
3294             [
3295                 cmap = UniGB-UTF16-H ,
3296                 BoldFont = FandolSong-Bold.otf
3297             ] { FandolSong-Regular.otf }
3298             \setCJKfamilyfont { zhhei }
3299             [
3300                 cmap = UniGB-UTF16-H ,
3301                 BoldFont = FandolHei-Bold.otf
3302             ] { FandolHei-Regular.otf }
3303             \setCJKfamilyfont { zhfs }
3304             [ cmap = UniGB-UTF16-H ] { FandolFang-Regular.otf }
3305             \setCJKfamilyfont { zhkai }
3306             [ cmap = UniGB-UTF16-H ] { FandolKai-Regular.otf }
3307             \ctex_punct_set:n { fandol }
3308             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3309             \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3310             \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3311             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3312             \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3313             \ctex_punct_map_bfseries:nn { \CJKsfdefault , zhhei } { zhheib }
3314         }
3315         {
3316             \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
3317             \ctex_punct_set:n { fandol }
3318             \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3319             \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3320             \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3321         }
3322     }
3323     { \ctex_fontset_error:n { fandol } }
3324 }

```

```

3324 }
3325 {
3326   \setCJKmainfont
3327     [ Extension = .otf , BoldFont = FandolSong-Bold , ItalicFont = FandolKai-Regular ]
3328     { FandolSong-Regular }
3329   \setCJKsansfont [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3330   \setCJKmonofont [ Extension = .otf ] { FandolFang-Regular }
3331   \setCJKfamilyfont { zhsong }
3332     [ Extension = .otf , BoldFont = FandolSong-Bold ] { FandolSong-Regular }
3333   \setCJKfamilyfont { zhhei }
3334     [ Extension = .otf , BoldFont = FandolHei-Bold ] { FandolHei-Regular }
3335   \setCJKfamilyfont { zhfs } [ Extension = .otf ] { FandolFang-Regular }
3336   \setCJKfamilyfont { zhkai } [ Extension = .otf ] { FandolKai-Regular }
3337 }
3338 </fandol>
3339 <*mac>

```

15.15.5 ctex-fontset-mac.def

OS X Mavericks (10.9) 预装的主要简体中文字体如下¹⁹:

```

/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}

```

¹⁹<http://support.apple.com/kb/HT5944>

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...
```

在 pdfTeX 下生成的 PDF 只有方框²⁰。华文细黑和华文仿宋的情况类似。

```
3340 \pdfutex_if_engine:TF
3341 { \ctex_fontset_error:n { mac } }
3342 {
3343   \setCJKmainfont [ BoldFont = STHeiti , ItalicFont = STKaiti ] { STSong }
3344   \setCJKsansfont [ BoldFont = STHeiti ] { STXihei }
3345   \setCJKmonofont { STFangsong }
3346   \setCJKfamilyfont { zhsong } { STSong }
3347   \setCJKfamilyfont { zhhei } { STHeiti }
3348   \setCJKfamilyfont { zhfs } { STFangsong }
3349   \setCJKfamilyfont { zhkai } { STKaiti }
3350 }

3351 

```

15.15.6 ctex-fontset-founder.def

```
3353 \pdfutex_if_engine:TF
3354 {
3355   \ctex_zhmap_case:nnn
3356   {
3357     \setCJKmainfont
3358     [ BoldFont = FZXBSK.TTF , ItalicFont = FZKTK.TTF ] { FZSSK.TTF }
3359     \setCJKsansfont [ BoldFont = FZHTK.TTF ] { FZXH1K.TTF }
3360     \setCJKmonofont { FZFSK.TTF }
3361     \setCJKfamilyfont { zhsong } [ BoldFont = FZXBSK.TTF ] { FZSSK.TTF }
3362     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
3363     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
3364     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
3365     \setCJKfamilyfont { zhli } { FZLSK.TTF }
3366     \setCJKfamilyfont { zhyou } [ BoldFont = FZY3K.TTF ] { FZY1K.TTF }
3367     \ctex_punct_set:n { founder }
3368     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3369     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
3370     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
3371     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3372     \ctex_punct_map_bfseries:nn { \CJKrmdefault , zhsong } { zhsongb }
3373     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
3374     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
3375   }
3376   {
3377     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
3378     \ctex_punct_set:n { founder }
3379     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3380     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3381     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3382   }
3383   { \ctex_fontset_error:n { founder } }
3384 }
3385 {
3386   \setCJKmainfont
3387   [ BoldFont = FZXiaoBiaoSong-B05 , ItalicFont = FZKai-Z03 ] { FZShuSong-Z01 }
```

²⁰<http://www.newsmth.net/bbscon.php?bid=460&id=312640>

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), X_YTeX 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。

```

3388     \setCJKsansfont [ BoldFont = FZHei-B01 ]
3389         { \xetex_if_engine:TF { 方正细黑一_GBK } { FZXH1K--GBK1-0 } }
3390     \setCJKmonofont { FZFangSong-Z02 }
3391     \setCJKfamilyfont { zhsong } [ BoldFont = FZXiaoBiaoSong-B05 ] { FZShuSong-Z01 }
3392     \setCJKfamilyfont { zhhei } { FZHei-B01 }
3393     \setCJKfamilyfont { zhkai } { FZKai-Z03 }
3394     \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
3395     \setCJKfamilyfont { zhli } { FZLiShu-S01 }
3396     \setCJKfamilyfont { zhyou } [ BoldFont = FZZhunYuan-M02 ] { FZXiYuan-M01 }
3397 }
3398 </founder>
3399 <*ubuntu>

```

15.15.7 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²¹。

```

3400 \pdfTeX_if_engine:TF
3401 {
3402     \ctex_zhmap_case:nnn
3403     {
3404         \setCJKmainfont
3405             [ BoldFont = wqy-zenhei.ttc , ItalicFont = ukai.ttc ] { uming.ttc }
3406         \setCJKsansfont { wqy-zenhei.ttc }
3407         \setCJKmonofont { uming.ttc }
3408         \setCJKfamilyfont { zhsong } { uming.ttc }
3409         \setCJKfamilyfont { zhhei } { wqy-zenhei.ttc }
3410         \setCJKfamilyfont { zhkai } { ukai.ttc }
3411         \ctex_punct_set:n { ubuntu }
3412         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3413         \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
3414         \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
3415         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3416         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3417     }
3418     {
3419         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
3420         \ctex_punct_set:n { ubuntu }
3421         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
3422         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
3423         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
3424     }
3425     { \ctex_fontset_error:n { ubuntu } }
3426 }
3427 {
3428     \setCJKmainfont [ ItalicFont = AR~PL~UKai~CN ] { AR~PL~UMing~CN }
3429     \setCJKsansfont { WenQuanYi~Zen~Hei }
3430     \setCJKmonofont { AR~PL~UMing~CN }
3431     \setCJKfamilyfont { zhsong } { AR~PL~UMing~CN }
3432     \setCJKfamilyfont { zhhei } { WenQuanYi~Zen~Hei }

```

²¹<http://www.freedesktop.org/wiki/Software/CJKUnifonts/Download/>

```

3433     \setCJKfamilyfont { zhkai } { AR~PL~UKai~CN }
3434   }
3435 </ubuntu>

```

15.15.8 中文字体命令

```

3436 <!*windows>
3437 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
3438 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
3439 <!ubuntu>\NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
3440 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
3441 <!*windowsnew|windowsold|founder>
3442 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
3443 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
3444 </windowsnew|windowsold|founder>
3445 <windowsnew>\NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
3446 </!windows>
3447 </fontset>
3448 <*zhmap>

```

15.15.9 zhmetrics 的字体映射

确认 `\catcode`, 没有重复载入检查。

```

3449 \begingroup\catcode61\catcode48\catcode32=10\relax%
3450 \catcode 35=6 % #
3451 \catcode123=1 % {
3452 \catcode125=2 % }
3453 \toks0{\endlinechar=\the\endlinechar\relax}%
3454 \toks2{\endlinechar=13 }%
3455 \def\x#1 #2 {%
3456   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
3457   \toks2\expandafter{\the\toks2 \catcode#1=#2 }%
3458 \x 13 5 % carriage return
3459 \x 32 10 % space
3460 \x 35 6 % #
3461 \x 40 12 % (
3462 \x 41 12 % )
3463 \x 45 12 % -
3464 \x 46 12 % .
3465 \x 47 12 % /
3466 \x 58 12 % :
3467 \x 60 12 % <
3468 \x 61 12 % =
3469 \x 64 11 % @
3470 \x 91 12 % [
3471 \x 93 12 % ]
3472 \x 123 1 % {
3473 \x 125 2 % }
3474 \edef\x#1{\endgroup%
3475   \edef\noexpand#1{%
3476     \the\toks0 %
3477     \let\noexpand\noexpand\noexpand#1%
3478     \noexpand\noexpand\noexpand\undefined%
3479     \noexpand\noexpand\noexpand\endinput}%
3480   \the\toks2}%
3481 \expandafter\x\cename ctex@zhmap@endinput\endcsname
3482 \input ifpdf.sty\relax

```

提供非 L^AT_EX 格式下的 `\ProvidesFile`。

```

3483 \begingroup
3484 \expandafter\ifx\cename ProvidesFile\endcsname\relax
3485 \long\def\x#1\ProvidesFile#2[#3]{%

```



```

3486     #1%
3487     \immediate\write-1{File: #2 #3}%
3488     \expandafter\xdef\csname ver@#2\endcsname{#3}}
3489     \expandafter\x%
3490 \fi
3491 \endgroup

```

15.15.9.1 zhwindowsfonts.tex

```

3492 <*windows>
3493 \ProvidesFile{zhwindowsfonts.tex}%
3494 [2014/06/03 v2.0 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]
3495
3496 \ifpdf
3497 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
3498 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
3499 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
3500 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
3501 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
3502 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
3503 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
3504
3505 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
3506 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
3507 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
3508 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
3509 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
3510 \pdfmapline{=unili@Unicode@ <simli.ttf}
3511 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
3512
3513 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
3514 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
3515 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
3516 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
3517 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
3518 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
3519
3520 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
3521 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
3522 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
3523 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
3524 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
3525 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
3526
3527 \else
3528 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
3529 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
3530 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
3531 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
3532 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
3533 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
3534 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
3535
3536 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
3537 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
3538 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
3539 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
3540 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
3541 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
3542 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
3543
3544 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -s .167 -v 50}
3545 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -s .167 -v 70}
3546 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -s .167 -v 150}
3547 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -s .167 -v 50}
3548 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -s .167 -v 150}
3549 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -s .167 -v 60}

```

```

3550
3551 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -s .167 -v 50}
3552 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -s .167 -v 70}
3553 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -s .167 -v 150}
3554 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -s .167 -v 50}
3555 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -s .167 -v 150}
3556 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -s .167 -v 60}
3557
3558 </windows>

```

15.15.9.2 zhadobefonts.tex

```

3559 <*adobe>
3560 \ProvidesFile{zhadobefonts.tex}%
3561 [2014/06/03 v2.0 Adobe font map loader for DVIPDFMx (CTEX)]
3562
3563 \ifpdf
3564 %% pdfTeX does not support OTF fonts
3565
3566 \else
3567 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3568 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3569 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3570 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3571 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3572
3573 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3574 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
3575 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
3576 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
3577 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
3578
3579 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3580 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3581 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3582 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3583
3584 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
3585 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
3586 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
3587 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
3588
3589 </adobe>

```

15.15.9.3 zhfandolfonts.tex

```

3590 <*fandol>
3591 \ProvidesFile{zhfandolfonts.tex}%
3592 [2014/06/03 v2.0 Fandol font map loader for DVIPDFMx (CTEX)]
3593
3594 \ifpdf
3595 %% pdfTeX does not support OTF fonts
3596
3597 \else
3598 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3599 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
3600 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
3601 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
3602 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
3603
3604 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3605 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
3606 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
3607 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
3608 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
3609
3610 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3611 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}

```

```

3612 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3613 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3614
3615 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
3616 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
3617 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
3618 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
3619
3620 (//fandol)

```

15.15.9.4 zhfounderfonts.tex

```

3621 (*founder)
3622 \ProvidesFile{zhfounderfonts.tex}%
3623 [2014/06/03 v2.0 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
3624
3625 \ifpdf
3626 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
3627 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
3628 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
3629 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
3630 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
3631 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
3632 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
3633
3634 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
3635 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
3636 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
3637 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
3638 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
3639 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
3640 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
3641
3642 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
3643 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
3644 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
3645 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
3646 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
3647 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
3648
3649 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
3650 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
3651 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
3652 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
3653 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
3654 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
3655
3656 \else
3657 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
3658 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
3659 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
3660 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
3661 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
3662 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
3663 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
3664
3665 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
3666 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
3667 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
3668 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
3669 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
3670 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
3671 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
3672
3673 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
3674 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
3675 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
3676 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}

```

```

3677 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
3678 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
3679
3680 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
3681 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
3682 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
3683 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
3684 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
3685 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
3686
3687 </founder>

```

15.15.9.5 zhubuntufonts.tex

```

3688 <*ubuntu>
3689 \ProvidesFile{zhubuntufonts.tex}%
3690 [2014/06/03 v2.0 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
3691
3692 \ifpdf
3693 \pdfmapline{=gbk@UGBK@ <uming.ttc}
3694 \pdfmapline{=gbksong@UGBK@ <uming.ttc}
3695 \pdfmapline{=gbkkai@UGBK@ <ukai.ttc}
3696 \pdfmapline{=gbkhei@UGBK@ <wqy-zenhei.ttc}
3697 \pdfmapline{=gbkfs@UGBK@ <uming.ttc}
3698 \pdfmapline{=gbkyou@UGBK@ <wqy-zenhei.ttc}
3699
3700 \pdfmapline{=cyberb@Unicode@ <uming.ttc}
3701 \pdfmapline{=unisong@Unicode@ <uming.ttc}
3702 \pdfmapline{=unikai@Unicode@ <ukai.ttc}
3703 \pdfmapline{=unihei@Unicode@ <wqy-zenhei.ttc}
3704 \pdfmapline{=unifs@Unicode@ <uming.ttc}
3705 \pdfmapline{=uniyou@Unicode@ <wqy-zenhei.ttc}
3706
3707 \pdfmapline{=gbksongsl@UGBK@ <uming.ttc}
3708 \pdfmapline{=gbkkaisl@UGBK@ <ukai.ttc}
3709 \pdfmapline{=gbkheisl@UGBK@ <wqy-zenhei.ttc}
3710 \pdfmapline{=gbkfssl@UGBK@ <uming.ttc}
3711 \pdfmapline{=gbkyousl@UGBK@ <wqy-zenhei.ttc}
3712
3713 \pdfmapline{=unisongsl@Unicode@ <uming.ttc}
3714 \pdfmapline{=unikaisl@Unicode@ <ukai.ttc}
3715 \pdfmapline{=uniheisl@Unicode@ <wqy-zenhei.ttc}
3716 \pdfmapline{=unifssl@Unicode@ <uming.ttc}
3717 \pdfmapline{=uniyousl@Unicode@ <wqy-zenhei.ttc}
3718
3719 \else
3720 \special{pdf:mapline gbk@UGBK@ unicode :0:uming.ttc}
3721 \special{pdf:mapline gbksong@UGBK@ unicode :0:uming.ttc}
3722 \special{pdf:mapline gbkkai@UGBK@ unicode :0:ukai.ttc}
3723 \special{pdf:mapline gbkhei@UGBK@ unicode :0:wqy-zenhei.ttc}
3724 \special{pdf:mapline gbkfs@UGBK@ unicode :0:uming.ttc}
3725
3726 \special{pdf:mapline cyberb@Unicode@ unicode :0:uming.ttc}
3727 \special{pdf:mapline unisong@Unicode@ unicode :0:uming.ttc}
3728 \special{pdf:mapline unikai@Unicode@ unicode :0:ukai.ttc}
3729 \special{pdf:mapline unihei@Unicode@ unicode :0:wqy-zenhei.ttc}
3730 \special{pdf:mapline unifs@Unicode@ unicode :0:uming.ttc}
3731
3732 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:uming.ttc -s .167}
3733 \special{pdf:mapline gbkkaisl@UGBK@ unicode :0:ukai.ttc -s .167}
3734 \special{pdf:mapline gbkheisl@UGBK@ unicode :0:wqy-zenhei.ttc -s .167}
3735 \special{pdf:mapline gbkfssl@UGBK@ unicode :0:uming.ttc -s .167}
3736
3737 \special{pdf:mapline unisongsl@Unicode@ unicode :0:uming.ttc -s .167}
3738 \special{pdf:mapline unikaisl@Unicode@ unicode :0:ukai.ttc -s .167}
3739 \special{pdf:mapline uniheisl@Unicode@ unicode :0:wqy-zenhei.ttc -s .167}
3740 \special{pdf:mapline unifssl@Unicode@ unicode :0:uming.ttc -s .167}
3741

```

```

3742 </ubuntu>
3743 \fi
3744
3745 \ctex@zhmap@endinput
3746 </zhmap>

```

15.15.10 制作 spa 文件

我们通过 Xe_{La}TeX 的 `\XeTeXglyphbounds` 取得字体中标点符号的边界信息, 为 CJKpunct 宏包制作 spa。

```

3747 <*spa>
3748 <*macro>
3749 \input expl3-generic %
3750 \ExplSyntaxOn
3751 \xetex_if_engine:F
3752 {
3753   \msg_new:nn { ctex } { xetex }
3754   { XeTeX~is~required~to~compile~this~document! }
3755   \msg_fatal:nn { ctex } { xetex }
3756 }

```

CJKpunct 定义的标点符号是:

```

‘ “ 「 『 [ ( [ { < « [ 【
—…、。、. . : ; ! ? % ] ) ] } > » ] 】 ’ ” 」 』

```

注意顺序不能改变。

```

3757 \seq_new:N \c__ctex_punct_seq
3758 \seq_gset_from_clist:Nn \c__ctex_punct_seq
3759 {
3760   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
3761   "3008 , "300A , "3016 , "3010 ,
3762   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
3763   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
3764   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
3765 }

```

`\ctex_make_spa:nn` #1 是 spa 文件名, #2 是由 CJK 族名与字体构成的逗号列表。

```

3766 \cs_new_protected_nopar:Npn \ctex_make_spa:nn #1#2
3767 {
3768   \iow_open:Nn \g__ctex_spa_iow {#1}
3769   \clist_map_inline:nn {#2}
3770   { \__ctex_write_family:nn ##1 }
3771   \iow_close:N \g__ctex_spa_iow
3772 }
3773 \iow_new:N \g__ctex_spa_iow
3774 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

```

(End definition for `\ctex_make_spa:nn`.)

```

3775 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
3776 {
3777   \group_begin:
3778   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
3779   \l__ctex_punct_font
3780   \clist_clear:N \l__ctex_punct_bounds_clist
3781   \seq_map_inline:Nn \c__ctex_punct_seq
3782   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \XeTeXcharglyph ##1 } }
3783   \iow_now:Nx \g__ctex_spa_iow
3784   {
3785     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

3786         { \l__ctex_punct_bounds_clist , , , }
3787     }
3788     \group_end:
3789 }
3790 \cs_new_protected_nopar:Npn \__ctex_save_bounds:n #1
3791 {
3792     \clist_put_right:Nx \l__ctex_punct_bounds_clist
3793     {
3794         \__ctex_calc_bounds:nn { \c_one } {#1} ,
3795         \__ctex_calc_bounds:nn { \c_three } {#1}
3796     }
3797 }
3798 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

3799 \cs_new_nopar:Npn \__ctex_calc_bounds:nn #1#2
3800 {
3801     \fp_eval:n
3802     {
3803         round
3804         (
3805             \dim_to_decimal_in_unit:nn
3806             { 100 \XeTeXglyphbounds #1 ~ #2 }
3807             { 1 em }
3808         )
3809     }
3810 }
3811 \ExplSyntaxOff
3812 \macro

```

下面是 C_TE_X 定义的一些字体。

```

3813 (*make)
3814 \input ctexspamacro %
3815
3816 \MAKESPA {ctexpunct.spa}
3817 {
3818     {adobezhsong}      {AdobeSongStd-Light} ,
3819     {adobezhhei}      {AdobeHeitiStd-Regular} ,
3820     {adobezhkai}      {AdobeKaitiStd-Regular} ,
3821     {adobezhfs}       {AdobeFangsongStd-Regular} ,
3822     {fandolzhsong}    {FandolSong} ,
3823     {fandolzhsongb}   {FandolSong-Bold} ,
3824     {fandolzhhei}     {FandolHei} ,
3825     {fandolzhheib}    {FandolHei-Bold} ,
3826     {fandolzhkai}     {FandolKai} ,
3827     {fandolzhfs}      {FandolFang} ,
3828     {founderzhsong}   {FZShuSong-Z01} ,
3829     {founderzhsongb}  {FZXiaoBiaoSong-B05} ,
3830     {founderzhhei}    {FZHei-B01} ,
3831     {founderzhheil}   {FZXiHeiI-Z08} ,
3832     {founderzhkai}    {FZKai-Z03} ,
3833     {founderzhfs}     {FZFangSong-Z02} ,
3834     {founderzhli}     {FZLiShu-S01} ,
3835     {founderzhyou}    {FZXiYuan-M01} ,
3836     {founderzhyoub}   {FZZhunYuan-M02} ,
3837     {ubuntuzhsong}    {AR PL UMinG CN} ,
3838     {ubuntuzhhei}     {WenQuanYi Zen Hei} ,
3839     {ubuntuzhkai}     {AR PL UKai CN} ,
3840     {windowszhsong}   {SimSun} ,
3841     {windowszhhei}    {SimHei} ,
3842     {windowszhkai}    {KaiTi} ,
3843     {windowszhfs}     {FangSong} ,
3844     {windowszhli}     {LiSu} ,
3845     {windowszhyou}    {YouYuan} ,
3846     {windowszhyahei}  {Microsoft YaHei} ,
3847     {windowszhyaheib} {Microsoft YaHei Bold}

```

```
3848 }  
3849  
3850 \primitive\end  
3851 </make>  
3852 </spa>
```

第 16 节 版本历史

v2.0		将标题汉化功能加入 <code>ctex.sty</code> 。	75
General: <code>c5size</code> , <code>cs4size</code> 是过时选项。	35	将中文字号功能提取到可以独立使用的 <code>ctexsize</code> 。	89
<code>captiondelimiter</code> 是过时选项。	93	默认关闭 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 的彩色设置。	68
<code>fancyhdr</code> 成为过时选项,原选项功能总是打开。	38	删除 <code>c19gbsn.fd</code> 和 <code>c19gkai.fd</code> 。	1
<code>fntef</code> 成为过时选项,原选项功能总是打开。	38	通过 <code>LuaTeX-ja</code> 宏包支持 <code>Lua\LaTeX</code> 。	45
<code>hyperref</code> 成为过时选项,原选项功能总是打开。	38	新增 <code>autoindent</code> 选项。	35
<code>indent</code> , <code>noindent</code> 是过时选项。	35	新增 <code>fontset</code> 选项。	36
<code>nofonts</code> , <code>adobefonts</code> , <code>winfonts</code> 是过时选项。	36	新增 <code>linespread</code> 选项。	35
<code>nopunct</code> 是过时选项。	37	新增 <code>scheme</code> 选项,并将 <code>cap</code> 和 <code>nocap</code> 列为过时选项 . . .	38
<code>nospace</code> 是过时选项。	37	新增 <code>zhmCJK</code> 支持选项。	36
<code>nozhmap</code> 是过时选项。	36	新增 <code>zihao</code> 选项。	35
<code>punct</code> 选项可以设置标点格式。	37	新增统一设置接口 <code>\ctexset</code> 。	40
<code>ctex</code> 宏包新增 <code>heading</code> 选项。	37	应用 <code>L\TeX3</code> 重新整理代码。	1
<code>\CTEXindent</code> , <code>\CTEXnoindent</code> 是过时命令。	74	中文字号不再采用近似值。	90
<code>\CTEXsetup</code> , <code>\CTEXoptions</code> 是过时命令。	15, 40	自动检测操作系统,载入对应的字体配置。	65
<code>\CTEXunderdot</code> , <code>\CTEXunderline</code> , <code>\CTEXunderdblline</code> ,		<code>\CJK@sur</code> : 解决与 <code>\nouppercase</code> 的冲突。	41
<code>\CTEXunderwave</code> , <code>\CTEXsout</code> , <code>\CTEXxout</code> 是过时命令;		<code>\end</code> : 解决 <code>etoolbox</code> 与 <code>breqn</code> 关于 <code>\end</code> 的冲突。	94
<code>CTEXfilltwosides</code> 是过时环境。	68	v2.0.1	
标题设置新增 <code>pagestyle</code> 选项。	78	General: 修复 <code>10pt</code> 、 <code>11pt</code> 等选项无效的问题。	38
调整 <code>\footnotesep</code> 的大小,以适合行距的变化。	93	v2.0.2	
兼容 <code>extsizes</code> 宏包、 <code>beamer</code> 、 <code>memoir</code> 等提供的更多字号选		General: 修复加载 <code>ctex</code> 宏包后章节标题后第一段无段首缩	
项。	38	进的问题	89

第 17 节 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	
\lfloor	2453, 2455, 2468, 2469, 2471
$\#$	160, 2427, 2950
<code>.../format+</code>	17
<code>.../nameformat+</code>	17
<code>.../numberformat+</code>	18
<code>.../titleformat+</code>	18
\surd	104, 918, 1281, 1282
\lt	103
\backslash	13, 14, 31, 104, 181, 222, 224, 226, 488, 678, 1076, 1158, 2053, 2675, 2676, 2753, 2754, 2997, 3027
$\{$	103, 1087, 1088
$\}$	1087, 1088
10pt	9, 38
11pt	9, 38
12pt	9, 38
A	
\A	105, 106
<code>\abovedisplayshortskip</code>	2844, 2852, 2863, 2883, 2891, 2902
<code>\abovedisplayskip</code>	2843, 2846, 2851, 2858, 2862, 2869, 2882, 2885, 2890, 2897, 2901, 2908
<code>\abstractname</code>	2044
<code>abstractname</code>	13
<code>\addCJKfontfeature</code>	1158, 1181
<code>\addCJKfontfeatures</code>	55, 1176, 1181
<code>\addcontentsline</code>	2180, 2181, 2183, 2205, 2206, 2208, 2250, 2254, 2257, 2332, 2346
<code>\addtocontents</code>	2260, 2261
<code>\addvspace</code>	2166, 2173, 2260, 2261
<code>adobefonts</code>	27
<code>AlternateFont</code>	23, 57
<code>\appendix</code>	82, 2402, 2403
<code>appendix/name</code>	21
<code>appendix/number</code>	21
<code>\appendixname</code>	2046, 2667
<code>appendixname</code>	13
<code>\apptocmd</code>	156
<code>\AtBeginDocument</code>	581, 1576, 2929
<code>\AtBeginDvi</code>	524
<code>\AtBeginShipoutFirst</code>	526, 1688
<code>\AtBeginUTFCommand</code>	722
<code>\AtEndOfClass</code>	212
<code>\AtEndOfPackage</code>	213
<code>\AtEndPreamble</code>	416, 592, 1608
<code>\AtEndUTFCommand</code>	728
<code>autoindent</code>	12, 35, 71

- B**
- \baselinestretch 1924
 - \begingroup 2324, 2359, 3449, 3483
 - \belowdisplayshortskip . 2845, 2853, 2864, 2884, 2892, 2903
 - \belowdisplayskip 2846, 2858, 2869, 2885, 2897, 2908
 - \bfdefault 939, 945, 1263, 1270
 - \bfseries 2189, 2192, 2215, 2218, 2227,
2239, 2282, 2287, 2305, 2514, 2519, 2526, 2532, 2536,
2539, 2547, 2554, 2561, 2572, 2576, 2583, 2596, 2609, 2615
 - \bibname 2048, 2053, 2056, 2057
 - bibname 13
 - bool commands:
 - \bool_gset_false:N 338, 343
 - \bool_gset_true:N 333
 - \bool_if:NT 1116
 - \bool_if:NTF 76, 2679
 - \bool_if:nTF 112
 - \bool_new:N 208, 1153
 - \bool_set_false:N 1128
 - \bool_set_true:N 1146
 - box commands:
 - \box_ht:N 2924
 - \box_new:N 27
 - \box_wd:N 1752, 2423
- C**
- c5size 26
 - cap 27
 - captiondelimiter 28, 93
 - \catcode 890, 3449, 3450, 3451, 3452, 3456, 3457
 - CCT 26
 - CCTfont 26
 - \ccwd 22, 68, 189, 261, 1752, 1756, 1758,
1788, 1789, 1875, 1913, 1932, 1933, 1940, 1967, 1968,
1969, 1972, 1974, 1975, 1986, 1997, 1998, 2944, 2945, 2946
 - \centering 2212, 2237, 2524, 2531, 2559, 2576
 - \chapter 2688
 - chapter/aftername 18
 - chapter/afterskip 20
 - chapter/beforeskip 19
 - chapter/format 17
 - chapter/indent 20
 - chapter/name 15
 - chapter/nameformat 17
 - chapter/number 16
 - chapter/numberformat 18
 - chapter/pagestyle 20
 - chapter/titleformat 18
 - \chaptermark 2259
 - \chaptername 2550
 - char commands:
 - \char_set_catcode:n 79, 554, 555
 - \char_set_catcode_active:N 1281
 - \char_set_catcode_letter:n 73
 - \char_set_catcode_math_superscript:n 549
 - \char_set_catcode_other:N 106, 160, 197, 198, 2427, 2950
 - \char_set_catcode_other:n 548, 731
 - \char_set_lccode:n 103, 104, 105, 735, 1282
 - \char_value_catcode:n 79, 554, 555
 - CharRange 23, 57
 - \Chinese 28, 2009
 - \chinese 22, 75, 2008, 2009, 2523, 2558
 - \CJK 1717
 - \CJKbold 3069, 3070, 3080, 3081, 3087, 3088, 3090, 3091, 3093,
3094, 3098, 3099, 3101, 3102, 3104, 3105, 3111, 3112,
3114, 3115, 3117, 3118, 3122, 3123, 3125, 3126, 3128, 3129
 - \CJKfamily . 53, 534, 536, 538, 540, 1013, 1035, 1214, 1216,
1218, 1220, 1366, 3437, 3438, 3439, 3440, 3442, 3443, 3445
 - \CJKfamilydefault . 529, 530, 540, 595, 1209, 1210, 1220,
1227, 1229, 1233, 1240, 1253, 1588, 1592, 1600, 1614, 1616
 - \CJKfilltwosides 1725
 - \CJKfontspec 55, 1171
 - \CJKglue 1751, 1762, 1776, 1785
 - \CJKhook 43, 610
 - \CJKnormal 3062, 3063, 3064, 3065, 3066, 3067, 3068,
3073, 3074, 3075, 3076, 3077, 3078, 3079, 3086, 3089,
3092, 3097, 3100, 3103, 3110, 3113, 3116, 3121, 3124, 3127
 - \CJKpunctmapfamily 643, 644, 645, 646, 647, 648, 652, 653, 654, 655, 656, 657
 - \CJKrmdefault . 521, 530, 531, 534, 1083, 1094, 1184, 1210,
1211, 1214, 1229, 1232, 1233, 1596, 3172, 3174, 3175,
3180, 3181, 3182, 3185, 3243, 3246, 3247, 3252, 3253,
3254, 3308, 3311, 3312, 3318, 3319, 3320, 3368, 3371,
3372, 3379, 3380, 3381, 3412, 3415, 3416, 3421, 3422, 3423
 - \CJKsfdefault 522, 532, 536, 1084, 1095, 1190, 1212, 1216, 1597,
3148, 3162, 3163, 3186, 3244, 3309, 3313, 3369, 3373, 3413
 - \CJKtilde 596
 - \CJKttdefault 523, 533, 538, 1085, 1096,
1195, 1213, 1218, 1598, 3173, 3187, 3245, 3310, 3370, 3414
 - clearalternatefont 24, 61
 - clist commands:
 - \clist_clear:N 960, 3780
 - \clist_concat:NNN 982
 - \clist_const:Nn 623
 - \clist_gput_right:Nn 437, 458, 460
 - \clist_gput_right:No 449
 - \clist_gset:Nn 1201
 - \clist_if_empty:NT 1288, 1544
 - \clist_if_empty:NTF 1304
 - \clist_map_break:n 2684
 - \clist_map_function:nN 1495, 1509
 - \clist_map_function:xN 1461, 1463
 - \clist_map_inline:Nn 613, 1397
 - \clist_map_inline:nn 422,
635, 1502, 1708, 1735, 2376, 2681, 2734, 2771, 2806, 3769
 - \clist_new:N 215, 979, 1202, 1339, 3798
 - \clist_put_left:Nx 983
 - \clist_put_right:Nn 1145
 - \clist_put_right:Nx 1385, 3792
 - \clist_set:Nn 1322, 1323

<code>\clist_set:No</code>	964	<code>\cs_set_eq:NN</code> .	172, 262, 267, 272, 541, 559, 579, 605, 609, 754, 801, 802, 954, 1013, 1221, 1358, 1366, 1376, 1377, 1378, 1450, 1451, 1452, 1591, 2015, 2020, 2023, 2028
<code>\contentsname</code>	2039	<code>\cs_set_nopar:cpx</code>	1745
<code>contentsname</code>	13	<code>\cs_set_protected_nopar:Npn</code>	604, 1762, 2833
cs commands:		<code>\cs_set_protected_nopar:Npx</code>	1792
<code>\cs:w</code>	2491	<code>\cs_to_str:N</code>	101, 1166
<code>\cs_end:</code>	2491	<code>\cs_undefine:c</code>	63, 992, 993, 997, 998
<code>\cs_generate_variant:Nn</code>	59, 65, 99, 153, 928, 1047, 1061, 1134, 1154, 1155, 1318, 1355, 1457, 1495, 1503	<code>\cs_undefine:N</code>	173, 1379
<code>\cs_gset_eq:cc</code>	1121, 1124	<code>cs4size</code>	26
<code>\cs_gset_eq:NN</code>	170, 234, 235, 236, 334, 339, 344, 396, 398, 432, 436, 791, 795, 2728	<code>\csname</code>	894, 896, 898, 914, 2313, 2314, 2329, 2331, 2334, 2336, 2337, 2344, 2345, 2348, 2350, 2351, 2957, 2960, 3481, 3484, 3488
<code>\cs_gset_nopar:Npx</code>	792	<code>\csuse</code>	2959, 2963
<code>\cs_gset_protected_nopar:cpx</code>	1009, 2380	<code>\CTeX</code>	23, 93, 2927, 2932
<code>\cs_gset_protected_nopar:Npn</code>	739, 2403	ctex commands:	
<code>\cs_gset_protected_nopar:Npx</code>	830, 1359, 1453	<code>\ctex_add_hook:cn</code>	30, 52
<code>\cs_if_exist:cF</code>	786, 828, 866	<code>\ctex_add_hook:Nn</code>	30, 54, 59, 610
<code>\cs_if_exist:CTF</code>	2478	<code>\ctex_add_to_selectfont:n</code>	70, 1790, 1815, 1901
<code>\cs_if_exist:Nf</code>	798, 1358, 1450, 1809, 2690	<code>\ctex_appto_cmd:NnTF</code>	32, 156, 1578
<code>\cs_if_exist:NTF</code>	2688	<code>\ctex_assign_heading_name:nn</code> ...	77, 2108, 2136, 2399
<code>\cs_if_exist_p:c</code>	112	<code>__ctex_assign_heading_name:nnn</code> ...	77, 2138, 2139
<code>\cs_if_exist_use:c</code>	1031, 1032	<code>\ctex_at_end:n</code>	34, 212, 213, 366, 1785, 3036
<code>\cs_if_exist_use:cF</code>	769	<code>\ctex_at_end_package:nn</code>	30, 30, 48, 525, 751, 756, 1682, 1684, 2437, 2458, 2502, 2955
<code>\cs_if_exist_use:cT</code>	62, 1472	<code>\ctex_auto_ignorespaces:</code>	43, 600, 609, 1832
<code>\cs_if_exist_use:CTF</code>	1053	<code>\l_ctex_autoindent_tl</code>	
<code>\cs_if_exist_use:NT</code>	2931	34, 209, 261, 266, 271, 1875, 1879, 1882, 1951, 1954
<code>\cs_if_free:cF</code>	615, 630, 1118	<code>__ctex_calc_bounds:nn</code>	3794, 3795, 3799
<code>\cs_if_free:CTF</code>	1262	<code>__ctex_ccglue:</code>	1776, 1785
<code>\cs_if_free:Nf</code>	753, 1902	<code>\l_ctex_ccglue_skip</code>	
<code>\cs_if_free:NTF</code>	1888	1763, 1767, 1770, 1772, 1780, 1925, 1936, 1945, 1981, 1989
<code>\cs_new:Npn</code>	151, 184, 191, 202, 956, 1034	<code>\ctex_CJK_input:n</code>	41, 544, 559
<code>\cs_new_eq:cc</code>	1744	<code>\c_ctex_class_tl</code>	
<code>\cs_new_eq:cN</code>	480, 481, 482, 493, 1730	90, 92, 463, 468, 473, 2684, 2686, 2692, 2694, 2695, 2697
<code>\cs_new_eq:Nc</code>	1796	<code>\ctex_declare_math_sizes:nnnn</code>	90, 2791, 2825
<code>\cs_new_eq:NN</code>	34, 36, 38, 207, 210, 211, 229, 600, 661, 807, 832, 1181, 1187, 1610, 1731, 1732, 1785, 1795, 2008, 2009, 2010, 2374, 2402, 2496, 2691, 3774	<code>__ctex_def_heading_keys:n</code>	77, 2102, 2150
<code>\cs_new_nopar:Npn</code>	847, 1026, 1079, 1090, 3799	<code>\ctex_default_pt:n</code>	33, 187, 191
<code>\cs_new_protected:Npn</code>	48, 54, 68, 100, 110, 123, 154, 155, 156, 157, 163, 177, 732, 957, 976, 1286, 1374, 1388, 1539, 1790, 2139, 2417, 2826, 2832, 3775	<code>__ctex_default_pt:w</code>	193, 202
<code>\cs_new_protected_nopar:cpx</code>	1714	<code>\ctex_detected_platform:</code> ..	65, 1619, 1637, 2977, 2986
<code>\cs_new_protected_nopar:Npn</code>	60, 83, 85, 212, 213, 519, 544, 561, 602, 608, 611, 628, 633, 641, 650, 659, 767, 776, 789, 794, 796, 808, 822, 849, 864, 877, 921, 934, 980, 988, 1007, 1027, 1037, 1062, 1100, 1112, 1114, 1136, 1138, 1223, 1248, 1257, 1302, 1313, 1340, 1349, 1356, 1362, 1382, 1395, 1414, 1428, 1435, 1447, 1467, 1483, 1501, 1504, 1506, 1519, 1531, 1565, 1581, 1586, 1619, 1637, 1656, 1667, 1671, 1721, 1727, 1748, 1759, 1787, 1887, 1889, 1914, 1949, 1959, 1964, 2076, 2102, 2372, 2474, 2483, 2488, 2744, 2759, 2791, 2796, 2969, 2973, 3766, 3790	<code>\l_ctex_encoding_tl</code>	33, 204, 205, 294, 295, 512, 515, 587, 595, 1678, 1860, 1864, 2006, 2723
<code>\cs_new_protected_nopar:Npx</code>	1167	<code>\c_ctex_engine_file_tl</code>	1643, 1653
<code>\cs_set_eq:cc</code>	617, 631	<code>\l_ctex_family_default_init_tl</code>	
<code>\cs_set_eq:cN</code>	1803	65, 1588, 1609, 1611, 1616
<code>\cs_set_eq:Nc</code>	2373	<code>__ctex_family_default_wrap:n</code>	1591, 1610, 1613
		<code>\ctex_file_input:n</code>	
		31, 83, 452, 662, 1861, 1866, 1867, 2697, 2724, 2725, 2835, 2837, 2978, 2982, 2990, 3036, 3135, 3136
		<code>\ctex_file_wrapper:nnn</code>	30, 68, 84, 87, 546
		<code>__ctex_fix_varioref_label:n</code>	84
		<code>\ctex_fix_varioref_label:n</code>	2486, 2488
		<code>\g_ctex_font_size_flag</code>	
		34, 229, 234, 235, 236, 432, 436, 457, 2727, 2728, 2834
		<code>\c_ctex_font_size_prop</code> 90, 2746, 2757, 2763, 2801, 2828	
		<code>\c_ctex_font_size_seq</code>	2755, 2758, 2769

`\l_ctex_font_size_tl` 2746, 2747, 2793, 2794, 2828, 2829
`\ctex_fontset_error:n`
 94, 2969, 3222, 3256, 3277, 3322, 3341, 3383, 3425
`\g_ctex_fontset_tl` ... 298, 1621, 1649, 1659, 1660,
 2975, 2978, 2981, 2982, 2988, 2990, 3010, 3012, 3017, 3034
`__ctex_get_font_sizes:Nn` 90, 2793, 2796
`\l_ctex_heading_bool` 377, 2679
`\c_ctex_headings_seq` 76, 2069, 2070, 2147, 2485
`\ctex_hypersetup:n`
 66, 66, 1667, 1671, 1674, 1677, 1680, 1695, 1698
`\ctex_if_autoindent_touched:F`
 34, 210, 262, 267, 272, 2731
`\ctex_if_ccglue_touched:` 1773
`\ctex_if_ccglue_touched:TF` 69, 1920
`\ctex_if_ccglue_touched_p:` 69
`\ctex_if_macosx:TF` 66, 1644, 1645, 1647, 1656
`\ctex_if_pdfmode:TF` 29, 36, 38, 1681, 3221, 3276
`\ctex_if_preamble:` 40
`\ctex_if_preamble:TF` 30, 3005
`\ctex_ignorespaces_case:N` .. 43, 602, 1829, 1832, 1835
`__ctex_initial_heading:n` 76, 2076, 2149
`\l_ctex_line_spread_fp` 93, 254, 2729, 2920, 2922
`\l_ctex_line_stretch_tl` 1907, 1912, 1913, 1917
`\ctex_load_fontset:` 94, 2973, 2994, 3013, 3035
`\ctex_load_zhmap:n` 528
`\ctex_load_zhmap:nmnn`
 41, 519, 3178, 3250, 3316, 3377, 3419
`\l_ctex_ltj_add_alternate_bool` 1116, 1128, 1146, 1153
`\ctex_ltj_add_font_features:n` 55, 1136, 1154
`\ctex_ltj_add_font_features:nm` 55, 1138, 1155
`\ctex_ltj_add_font_features:x` 1178
`\ctex_ltj_add_font_features:xn` 1137
`__ctex_ltj_alternate_cs:n`
 993, 996, 997, 998, 1031, 1032, 1034, 1119,
 1122, 1123, 1125, 1126, 1308, 1345, 1407, 1432, 1472, 1489
`\l_ctex_ltj_alternate_family_tl` 1370, 1373
`\l_ctex_ltj_alternate_options_clist`
 1323, 1324, 1335, 1339
`\l_ctex_ltj_alternate_seq`
 961, 1295, 1296, 1301, 1307, 1315, 1563
`\l_ctex_ltj_base_CJKfamily_tl`
 962, 978, 1029, 1407, 1432
`\l_ctex_ltj_base_family_tl` 61,
 994, 1024, 1030, 1351, 1392, 1451, 1470, 1475, 1476, 1488
`__ctex_ltj_change_encoding:` 51, 934, 1012, 1365
`\l_ctex_ltj_char_range_clist` 960,
 1277, 1288, 1304, 1322, 1334, 1346, 1369, 1391, 1542, 1544
`\ctex_ltj_char_range_key:nm` 63, 1536, 1539
`__ctex_ltj_char_range_parse_feature:w`
 63, 1547, 1551
`\g_ctex_ltj_char_range_prop` ... 62, 1399, 1510, 1515
`\l_ctex_ltj_char_range_tl`
 1399, 1402, 1508, 1510, 1512, 1514, 1521
`__ctex_ltj_check_family:n` 52, 988, 1306
`\ctex_ltj_clear_alternate_font:n` 1461, 1467
`\l_ctex_ltj_current_font_tl`
 47, 769, 773, 774, 786, 828, 833
`\l_ctex_ltj_current_shape_tl` 851, 856, 861, 862
`\ctex_ltj_declare_alternate_shape:nmnnnn`
 60, 1368, 1388, 1390
`\ctex_ltj_declare_char_range:n` 1501, 1503
`__ctex_ltj_declare_char_range:nm` 1502, 1504
`\ctex_ltj_declare_char_range:nm` 62, 1505, 1506
`\ctex_ltj_declare_char_range:x` 1498
`\ctex_ltj_def_char_range_key:n` 63, 1511, 1531
`\g_ctex_ltj_default_features_clist` . 982, 1201, 1202
`\ctex_ltj_ensure_default_family:` ... 56, 1223, 1605
`\ctex_ltj_extract_font:` 48, 801, 808
`__ctex_ltj_family_csname:n` ... 992, 1009, 1026, 1053
`\g_ctex_ltj_family_font_name_prop`
 52, 967, 986, 990, 1064, 1140, 1225, 1237
`\g_ctex_ltj_family_font_options_prop`
 52, 968, 987, 1143
`\ctex_ltj_family_if_exist:nTF` 53
`\ctex_ltj_family_if_exist:xN` 1048
`\ctex_ltj_family_if_exist:xNF` 1227
`\ctex_ltj_family_if_exist:xNT` 1253
`\ctex_ltj_family_if_exist:xNTF` 1039, 1232, 1250, 1470
`\g_ctex_ltj_family_int` . 1106, 1108, 1384, 1385, 1387
`\g_ctex_ltj_family_name_prop`
 52, 985, 994, 1016, 1050, 1351
`\l_ctex_ltj_family_tl`
 1041, 1046, 1103, 1104, 1119, 1123, 1126, 1137, 1464, 1465
`__ctex_ltj_family_unknown_warning:n`
 53, 1044, 1062, 1480
`\l_ctex_ltj_font_options_clist` 964,
 969, 970, 971, 979, 1015, 1144, 1145, 1148, 1342, 1343, 1347
`\ctex_ltj_fontspec:nm` 54, 1100, 1113
`__ctex_ltj_fontspec:nm` 1114, 1134
`__ctex_ltj_fontspec:xnm` 1107
`\ctex_ltj_fontspec:xx` 1112, 1147, 1173
`\g_ctex_ltj_fontspec_family_tl`
 1014, 1017, 1018, 1019, 1025, 1030, 1055
`\g_ctex_ltj_fontspec_prop` 1102, 1130, 1135
`__ctex_ltj_gset_family_cs:nm` 52, 972, 1007
`\ctex_ltj_if_alternate_shape_exist:n` 842
`\ctex_ltj_if_alternate_shape_exist:nF` 824
`\ctex_ltj_if_alternate_shape_exist:nT` 811, 851
`\ctex_ltj_if_alternate_shape_exist:nTF` 49
`\l_ctex_ltj_jfm_tl` 51, 924, 925, 927, 938, 940, 983
`\c_ctex_ltj_math_fam_int` 946, 947
`\c_ctex_ltj_math_family_tl`
 1259, 1261, 1263, 1266, 1270
`\c_ctex_ltj_math_tl`
 941, 942, 944, 946, 1199, 1250, 1260, 1265, 1269
`__ctex_ltj_msg_def_family_map:n` 1077, 1079
`__ctex_ltj_msg_family_map:n` 1006, 1076, 1090
`__ctex_ltj_noembed_wrap:n` 954, 956
`__ctex_ltj_patch_external_font:w` 49, 814, 847
`\ctex_ltj_pickup_font:` 47, 782, 796, 807, 871
`__ctex_ltj_pop_fontname:` 785, 794, 873

- _ctex_ltj_push_fontname:n 781, 789, 870
- \ctex_ltj_reset_alternate_font:n 1463, 1483
- \g_ctex_ltj_reset_alternate_prop
..... 999, 1474, 1486, 1494
- \ctex_ltj_save_alternate_family:Nnnn
..... 59, 1333, 1353, 1356
- \ctex_ltj_save_alternate_seq:cn 1308
- \ctex_ltj_save_alternate_seq:Nn 58, 1313, 1318
- \ctex_ltj_save_alternate_seq:Nnnwn 58
- \ctex_ltj_save_alternate_seq:Nnnwnw 1316, 1319
- _ctex_ltj_save_alternate_shape:cmn ... 1406, 1431
- _ctex_ltj_save_alternate_shape:Nnn 61, 1447, 1457
- \ctex_ltj_save_char_range:n 62, 1509, 1516
- \ctex_ltj_save_char_range:nn 1518, 1519
- _ctex_ltj_save_fontname:w 791, 795
- \ctex_ltj_select_alternate_font: 49, 849, 1818
- \ctex_ltj_select_font: 47, 767, 1817
- _ctex_ltj_select_font_aux: 47, 770, 776, 787
- \ctex_ltj_set_alternate_family:coonn 1344
- _ctex_ltj_set_alternate_family:n 1020, 1027
- \ctex_ltj_set_alternate_family:nn .. 59, 1311, 1340
- \ctex_ltj_set_alternate_family:nnF ... 58, 965, 1302
- \ctex_ltj_set_alternate_family:nnn
..... 59, 1352, 1360, 1362
- \ctex_ltj_set_alternate_family:Nnnnn ... 1349, 1355
- \ctex_ltj_set_alternate_seq:n 57, 1275, 1286
- \ctex_ltj_set_alternate_shape:n ... 1416, 1428, 1439
- \ctex_ltj_set_alternate_shape:nnN .. 61, 1401, 1435
- \ctex_ltj_set_alternate_shape:nnn .. 60, 1404, 1411
- \ctex_ltj_set_alternate_shape:nnnn 1413, 1414
- \ctex_ltj_set_alternate_shape:Nnnnnnn 60, 1391, 1395
- \ctex_ltj_set_family:nnn 51, 957, 977, 1131
- \ctex_ltj_set_family:xxx
..... 976, 1162, 1169, 1184, 1190, 1195, 1199
- \ctex_ltj_subst_font: 48, 802, 822
- \ctex_ltj_swap_cs:NN 59, 1367, 1374
- \ctex_ltj_switch_family:n 1037, 1047, 1132, 1168
- \ctex_ltj_switch_family:x 1036, 1104
- _ctex_ltj_tmp:w 1376, 1378, 1379
- \l_ctex_ltj_tmp_seq 1292, 1293, 1296, 1300
- \l_ctex_ltj_tmp_tl 963, 964, 990, 1001, 1004,
1141, 1149, 1165, 1168, 1169, 1250, 1251, 1253, 1254,
1290, 1291, 1292, 1326, 1328, 1329, 1336, 1546, 1554, 1563
- \l_ctex_ltj_tmpa_tl 1227, 1232
- _ctex_ltj_um_char:Nw 732, 754
- \g_ctex_ltj_unknown_family_seq ... 1066, 1068, 1073
- _ctex_ltj_update_family_uid:N . 60, 970, 1342, 1382
- \ctex_ltj_update_mathfont: 57, 1245, 1248
- \ctex_ltj_update_mathfont:n 1251, 1254, 1257
- \ctex_ltj_update_xkanjiskip: 64, 1565, 1819
- _ctex_ltj_use_global_options:N 971, 980, 1324, 1343
- \l_ctex_ltj_xkanjiskip
..... 64, 1568, 1570, 1571, 1574, 1575
- \ctex_lua_now_x:n 29, 34,
725, 743, 818, 844, 848, 853, 858, 880, 933, 1430, 1454, 1623
- \c_ctex_macosx_file_tl 1658, 1662
- \ctex_make_spa:nn 109, 3766, 3774
- \ctex_mono_jfm:n 51, 921, 928
- \ctex_mono_jfm:o 929, 1852
- \ctex_package_end_hook:n 30, 60, 65
- \ctex_package_end_hook:o 30, 67
- \ctex_parse_name:NN 31, 100, 154, 155, 156
- \ctex_parse_name:NNn 110, 153
- _ctex_parse_name:nNNnN 117, 123
- \ctex_parse_name:NNx 101
- _ctex_parse_name:w 136, 142, 151
- _ctex_patch_cmd:Nnn 161, 163
- \ctex_patch_cmd:Nnn 32, 157, 2446,
2448, 2452, 2455, 2461, 2463, 2467, 2470, 2944, 2945, 2946
- \ctex_patch_cmd:NnnTF 32, 154, 168, 2957
- \ctex_patch_failure:N 175, 177, 535, 537, 539,
542, 1215, 1217, 1219, 1222, 1579, 1805, 2433, 2440, 2964
- \ctex_plane_to_utfxvibe:Nn 41, 561, 579
- \ctex_preto_cmd:NnTF 32, 155, 534, 536,
538, 540, 1214, 1216, 1218, 1220, 1800, 2432, 2439, 2963
- \l_ctex_punct_bounds_clist ... 3780, 3786, 3792, 3798
- \c_ctex_punct_family_clist 613, 623
- \l_ctex_punct_font 3778, 3779
- \ctex_punct_map_bfseries:nn
..... 44, 633, 3163, 3175, 3181, 3247,
3253, 3312, 3313, 3319, 3372, 3373, 3374, 3380, 3416, 3422
- \ctex_punct_map_family:nn ... 44, 628, 3148, 3162,
3172, 3173, 3180, 3243, 3244, 3245, 3252, 3308, 3309,
3310, 3318, 3368, 3369, 3370, 3379, 3412, 3413, 3414, 3421
- \ctex_punct_map_itshape:nn 44, 650,
3174, 3182, 3246, 3254, 3311, 3320, 3371, 3381, 3415, 3423
- \ctex_punct_map_series:nnn 637, 638, 641
- \c_ctex_punct_seq 3757, 3758, 3781
- \ctex_punct_set:n 43, 611,
3143, 3179, 3242, 3251, 3307, 3317, 3367, 3378, 3411, 3420
- \ctex_punct_space:nn 44, 659, 661
- \l_ctex_punct_tl
..... 353, 583, 584, 670, 929, 1849, 1850, 1851, 1852
- _ctex_save_bounds:n 3782, 3790
- _ctex_save_font_size:nn 90, 2759, 2790
- _ctex_save_selectfont: 1796, 1803
- \ctex_scheme_input:n 31, 85, 99
- \ctex_scheme_input:o 2504, 2699
- \l_ctex_scheme_tl 378, 2504, 2699
- \g_ctex_section_depth_flag
..... 33, 207, 396, 398, 2618, 2638, 2658
- \ctex_select_size:.. 1876, 1883, 1886, 1887, 1908, 1962
- \ctex_set_default_ccwd:Nn ... 33, 184, 271, 1882, 1907
- \ctex_set_font_size:Nnn 91, 2826, 2841,
2849, 2860, 2871, 2872, 2873, 2874, 2875, 2876, 2877,
2880, 2888, 2899, 2910, 2911, 2912, 2913, 2914, 2915, 2916
- _ctex_set_font_size:nnNn 2829, 2832
- \ctex_set_ignorespaces: 43, 604, 606, 608, 610
- \g_ctex_spa_iow 3768, 3771, 3773, 3783
- \g_ctex_std_options_clist
..... 34, 215, 437, 449, 458, 460, 464, 469, 474
- _ctex_tmp:w 170, 172, 173

- \l__ctex_tmp_box 27, 29, 1751, 1752, 2419, 2423
 - \l__ctex_tmp_dim 26, 29, 1917, 1918, 1923, 1926, 1932, 1971, 1973, 1975, 1984, 1986
 - \l__ctex_tmp_int 25, 29, 564, 566, 567, 569, 573, 575, 1931, 1933, 1934, 1940, 1941, 1973, 1974, 1975, 1978, 1979, 1984, 1986, 2375, 2379, 2383
 - \l__ctex_tmp_tl . 24, 29, 421, 428, 443, 444, 478, 480, 481, 482, 493, 563, 565, 2104, 2146, 2152, 2153, 2801, 2802
 - \ctex_update_ccglue: 69, 1759, 1927, 1946, 1990
 - \ctex_update_ccwd: 68, 1748, 1921, 1998
 - \ctex_update_default_family: 64, 1586, 1608
 - \ctex_update_em_unit: 69, 1787, 1916, 1966
 - \ctex_update_parindent: 73, 1894, 1949, 1991
 - \ctex_update_size: 72, 1888, 1889, 1902
 - \ctex_update_stretch: 72, 1893, 1914
 - \ctex_update_ziju: 73, 1896, 1964
 - \ctex_varioref_hook: 84, 2483, 2494, 2502
 - __ctex_write_family:nn 3770, 3775
 - \c__ctex_zero_tl 1891, 1898, 1900
 - \ctex_zhmap_case:nnn 34, 211, 334, 339, 344, 3141, 3224, 3279, 3355, 3402
 - \g__ctex_zhmCJK_bool ... 34, 208, 333, 338, 343, 511, 586
 - \ctex_zihao:n 89, 2743, 2744
 - \ctex_ziju:n 1958, 1959
 - \l__ctex_ziju_dim 1967, 1968, 1972, 1983, 1985, 1989, 1993
 - \l__ctex_ziju_tl 1891, 1899, 1900, 1961, 1967
 - \ctexbibname 2054, 2063
 - \CTEXcounter 2010
 - \CTEXdigits 23, 75, 2013
 - \CTEXfilltwosides 27, 1721, 1731
 - \CTEXindent 27, 74, 1994, 1996
 - \CTEXnoindent 27, 74, 2000, 2002
 - \CTEXnumber 23, 75, 2011
 - \CTEXoptions 28, 40, 503, 505
 - \ctexset 6, 40, 495, 500, 506
 - \CTEXsetfont 22, 72, 1886
 - \CTEXsetup 27, 40, 497, 499
 - \CTEXsout 27
 - \ctexspadef 44, 661, 3785
 - \CTEXthechapter 16, 2249, 2252, 2453, 2468, 2469
 - \CTEXtheparagraph 16
 - \CTEXthepart 16, 2181, 2206
 - \CTEXthesection 16, 2446, 2455, 2462, 2471
 - \CTEXthesubparagraph 16
 - \CTEXthesubsection 16, 2448, 2464
 - \CTEXthesubsubsection 16
 - \CTEXunderdblline 27
 - \CTEXunderdot 27
 - \CTEXunderline 27
 - \CTEXunderwave 27
 - \CTEXxout 27
 - \CurrentOption 449
- D**
- declarecharrange 24, 62
 - \DeclareFontEncoding 931
 - \DeclareFontFamily . 936, 3054, 3055, 3056, 3057, 3058, 3059
 - \DeclareFontShape 937, 939, 3062, 3063, 3064, 3065, 3066, 3067, 3068, 3069, 3070, 3073, 3074, 3075, 3076, 3077, 3078, 3079, 3080, 3081, 3086, 3087, 3088, 3089, 3090, 3091, 3092, 3093, 3094, 3097, 3098, 3099, 3100, 3101, 3102, 3103, 3104, 3105, 3110, 3111, 3112, 3113, 3114, 3115, 3116, 3117, 3118, 3121, 3122, 3123, 3124, 3125, 3126, 3127, 3128, 3129
 - \DeclareFontSubstitution 932
 - \DeclareMathSizes 2794
 - \DeclareSymbolFont 942, 1260
 - \def 887, 893, 906, 912, 2177, 2201, 2223, 2235, 2244, 2268, 2292, 2311, 2315, 2341, 2356, 2367, 2368, 2854, 2865, 2893, 2904, 3455, 3485
 - \defaultCJKfontfeatures 56, 1200, 1208
 - dim commands:
 - \dim_abs:n 1985
 - \dim_add:Nn 1968
 - \dim_compare:nNnF 1953, 1976
 - \dim_compare:nNnTF 186, 1918, 1969
 - \dim_eval:n 1938, 1984
 - \dim_max:nn 2422
 - \dim_min:nn 1985
 - \dim_new:N 26, 1758, 1993
 - \dim_set:Nn 1752, 1788, 1789, 1917, 1923, 1954, 1967, 1971, 1998, 2420, 2924
 - \dim_sub:Nn 1975
 - \dim_to_decimal:n 33, 2765, 2766, 2803
 - \dim_to_decimal_in_unit:nn 3805
 - \dim_use:N 194, 2924
 - \dim_zero:N 2004
 - \directlua 708, 709, 888
- E**
- \E 106
 - \edef 900, 3474, 3475
 - \else 909, 914, 916, 2170, 2182, 2207, 2253, 2256, 2264, 2274, 2298, 2318, 2333, 2340, 2347, 2366, 3527, 3566, 3597, 3656, 3719
 - else commands:
 - \else: 37, 44, 514, 845, 1777, 2928
 - \end 94, 2957, 2963, 2964, 3850
 - \endCJKfilltwosides 1727
 - \endcsname 894, 896, 898, 914, 2313, 2314, 2329, 2331, 2334, 2336, 2337, 2344, 2345, 2348, 2350, 2351, 2957, 2960, 3481, 3484, 3488
 - \endCTEXfilltwosides 1727, 1732
 - \endgroup 2330, 2365, 3474, 3491
 - \endinput 3479
 - \endlinechar 3453, 3454
 - etex commands:
 - \etex_dimexpr:D 194, 1932
 - \etex_glueexpr:D 1924
 - \EverySelectfont 1809, 1812

- exp commands:
- \exp_after:wN 193, 798, 814, 816, 1651, 1653, 1654, 2747, 2794, 2829
 - \exp_args:cc 2479
 - \exp_args:Nc 126
 - \exp_args:Nnc 84
 - \exp_args:NNf 1554
 - \exp_args:NNNo 2923
 - \exp_args:Nno 2491
 - \exp_args:No 565, 3782
 - \exp_args:Nx 366, 592, 666, 1958, 2743
 - \exp_last_unbraced:Nf 136
 - \exp_not:c 118, 833, 1716, 1718, 1745, 2082, 2083, 2084, 2088, 2089, 2090, 2091, 2094, 2095, 2098, 2099, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2119, 2121, 2123, 2125, 2127, 2384, 2385, 2386, 2389
 - \exp_not:N 118, 594, 595, 596, 760, 761, 763, 869, 1016, 1017, 1018, 1019, 1333, 1454, 1596, 1597, 1598, 1613, 1717, 2090, 2382, 2389, 2763, 2988
 - \exp_not:n 75, 134, 1013, 1014, 1591
 - \exp_not:o 443, 762, 764, 1015, 1148, 1149, 1334, 1335, 1336, 1360, 1454, 1614, 1793, 2152, 2480
 - \expandafter 914, 915, 917, 3456, 3457, 3481, 3484, 3488, 3489
 - \ExplSyntaxOff .. 72, 78, 167, 690, 886, 1799, 2431, 2954, 3811
 - \ExplSyntaxOn 77, 719, 920, 1807, 2443, 2967, 3750
- F**
- \familydefault 1594
 - fancyhdr 27, 38
 - \fangsong 3439
 - \fi 909, 915, 916, 917, 2164, 2172, 2184, 2191, 2209, 2217, 2255, 2258, 2267, 2276, 2284, 2285, 2300, 2321, 2338, 2352, 2354, 2369, 2468, 3490, 3743
- fi commands:
- \fi: 39, 46, 461, 543, 580, 598, 845, 1777, 2449, 2456, 2838, 2928
 - \figurename 2042
 - figurename 13
- file commands:
- \g_file_current_name_tl 682, 1654
 - \file_if_exist:nTF 1639, 1641, 1643, 1658, 2981, 3134, 3151
 - \file_if_exist_input:nF 92
 - \file_input:n 84, 93, 95, 524, 526, 552
- five commands:
- \c_five 2766
 - fntef 27, 38
 - \font 3054, 3055, 3056, 3057, 3058, 3059
- font@name commands:
- \font@name: 903
- fontset 8, 36, 95
- \fontsize 2747
- fontspec commands:
- \g_fontspec_encoding_tl 935
 - \l_fontspec_family_tl 1393
 - _fontspec_namewrap:n 51, 954
 - \fontspec_set_family:Nnn 1014, 1370
 - \footnotesep 93, 2924
 - \footnotesize 2860, 2899, 2923
- four commands:
- \c_four 398, 573, 575
- fp commands:
- \fp_compare:nNnF 2920
 - \fp_compare:nNnT 2729
 - \fp_eval:n 1961, 3801
 - \fp_use:N 1898, 2922
- G**
- GBK 7, 36
 - \gdef 891, 2407, 2408, 2409, 2412, 2413, 2414
 - \globaljfont 816
- group commands:
- \group_begin: 102, 114, 159, 166, 196, 724, 730, 734, 778, 800, 826, 868, 959, 1011, 1280, 1364, 1449, 1469, 1485, 1590, 2093, 2426, 2923, 2949, 3777
 - \group_end: 109, 125, 165, 171, 175, 201, 728, 738, 750, 783, 804, 839, 872, 974, 1021, 1285, 1371, 1455, 1481, 1492, 1602, 2096, 2430, 2923, 2953, 3788
 - \group_insert_after:N 836, 837
- H**
- hbox commands:
- \hbox_set:Nn 1751, 2419
- heading 10, 37
- \heiti 3438
 - \hskip 2326, 2342, 2361, 2367, 2368, 2462, 2464
 - \hspace 2180, 2181, 2205, 2206, 2252
 - \Huge 2218, 2239, 2287, 2305, 2536, 2554, 2877, 2916
 - \huge 2192, 2215, 2227, 2282, 2519, 2532, 2539, 2547, 2561, 2876, 2915
- hyperref 27, 38
- \hypersetup 1668
 - \hyphenchar 3054, 3055, 3056, 3057, 3058, 3059
- I**
- if commands:
- \if_bool:N 511, 586
 - \if_case:w 457, 2834
 - \if_cs_exist:N 560
 - \if_meaning:w 42, 1776
 - \ifdim 2168, 2272, 2296, 2323, 2358
 - \ifincsname 2928
 - \ifmmode 912
 - \IfNoValueF 501, 507
 - \IfNoValueTF 1166, 1418, 1523, 1556, 1558, 2142
 - \ifnum ... 2178, 2188, 2203, 2214, 2245, 2280, 2316, 2333, 2347
 - \ifpdf 35, 3496, 3563, 3594, 3625, 3692
 - \ifx 914, 3484
 - \immediate 3487
 - indent 26, 35
 - \indexname 2045
 - indexname 13
 - \input 3482, 3749, 3814

int commands:		linespread	11, 35
\int_compare:nNnF	1978, 2727	linestretch	12, 72
\int_compare:nNnTF	566, 1934, 2618, 2638, 2658	\linewidth	1932, 1940, 1972
\int_const:Nn	946	\lishu	3442
\int_div_truncate:nn	72, 573	\listfigurename	2040
\int_eval:n	1419, 1421, 1422, 1524, 1526, 1527	listfigurename	13
\int_from_hex:n	565	\listtablename	2041
\int_gincr:N	1106, 1384	listtablename	13
\int_incr:N	2379	\LoadClass	465, 470, 475
\int_mod:nn	575	\long	3485
\int_new:N	25, 1387	\lstlistingname	761, 762
\int_set:Nn	556, 564, 1931, 1973	\lstlistlistingname	763, 764
\int_set_eq:NN	550	\ltjdefcharrange	691, 692, 693, 695, 699, 700, 703, 706
\int_sub:Nn	569, 1979	LTJFONTUID	60
\int_to_Hex:n	567, 572, 574	\ltjgetparameter	1568, 1756, 1780
\int_use:N	556, 1108, 1385, 2383, 3782	\ltjsetkanjiskip	1770
\int_zero:N	2375	\ltjsetmathletter	748
\interlinepenalty	2185, 2211, 2224, 2236, 2286, 2304, 2327, 2328, 2362, 2363	\ltjsetparameter	707, 713, 1582
iow commands:		\ltjsetxkanjiskip	1571
\iow_close:N	3771	lua commands:	
\iow_new:N	3773	\lua_now_x:n	29
\iow_now:Nx	3783	luatex commands:	
\iow_open:Nn	3768	\luatex_directlua:D	34
\itemsep	2857, 2868, 2896, 2907	\luatexluaescapestring	844, 856, 883, 889, 1409, 1424, 1425, 1442, 1443, 1452
J			
\jfam	916, 947	M	
K			
\kaishu	3440	\MAKESPA	3774, 3816
kernel commands:		\markboth	2194, 2210
\l_kernel_expl_bool	76	\mathgroup	916
keys commands:		max commands:	
\l_keys_choice_tl	367	\c_max_dim	1918
\keys_define:nn	230, 443, 446, 951, 1273, 1381, 1458, 1496, 1535, 1824, 1845, 1869, 1903, 2016, 2037, 2057, 2060, 2063, 2152, 2155, 2395, 2397, 2934, 3001	\mddefault	932, 937, 943, 1261, 1266
\keys_if_exist:nnF	1533	minus commands:	
\keys_if_exist:nnT	2714, 2717	\c_minus_one	229, 550, 2727
\l_keys_key_tl	218, 220, 222	mode commands:	
\keys_set:nn	25, 244, 250, 306, 315, 324, 350, 360, 367, 374, 385, 391, 496, 501, 507, 1713, 2507, 2544, 2566, 2580, 2593, 2606, 2612, 2620, 2632, 2640, 2652, 2659, 2660, 2661, 2702, 2716, 2718, 2726, 2730, 2732, 3043	\mode_if_math:TF	741
\keys_set_known:nnN	963	msg commands:	
L			
\labelformat	2491	\msg_critical:nnn	2970
\LARGE	2875, 2914	\msg_critical:nxx	682
\Large	2189, 2514, 2526, 2572, 2576, 2874, 2913	\msg_error:nn	492, 3021
\large	2583, 2873, 2912	\msg_error:nnn	18, 20, 673, 686, 689, 2748, 2830
\leavevmode	2164	\msg_error:nxx	2032
\leftmargin	2854, 2865, 2893, 2904	\msg_error:nxxx	2987, 3016
\leftmargini	2854, 2865, 2893, 2904	\msg_fatal:nn	3755
\let	892, 895, 897, 908, 909, 918, 2317, 2847, 2886, 3477	\msg_new:nn	3753
\linespread	2922	\msg_new:nnn	179, 217, 219, 221, 223, 225, 676, 1005, 1074, 1156, 1704, 2051, 2673, 2971, 3030
		\msg_new:nnnn	10, 28, 485, 2034, 2750, 2995, 3024
		\msg_warning:nn	1151, 1706, 1842, 2062
		\msg_warning:nnn	242, 248, 276, 286, 301, 310, 319, 348, 358, 372, 383, 389, 403, 408, 414, 1069, 2938, 3008
		\msg_warning:nnnn	499, 505, 1716, 1723, 1996, 2002
		\msg_warning:nxx	178, 2695
		\msg_warning:nxxx	1001

N	
nan commands:	
\ <code>\c_nan_fp</code>	255
\ <code>\newCJKfontfamily</code>	55, 1163
\ <code>\NewDocumentCommand</code>	495, 497, 503, 1035, 1161, 1163, 1171, 1176, 1182, 1188, 1193, 1198, 1200, 1319, 1411, 1516, 1551, 1886, 1957, 1994, 2000, 2011, 2013, 2136, 2742, 2927, 3437, 3438, 3439, 3440, 3442, 3443, 3445
\ <code>\newfontfeature</code>	948, 949, 950
nil commands:	
\ <code>\q_nil</code>	732
\ <code>\nobreak</code>	2189, 2195, 2229, 2282, 2287, 2288, 2305, 2306, 2518, 2553
\ <code>\nobreakspace</code>	2189, 2215
nocap	27
\ <code>\noexpand</code>	3475, 3477, 3478, 3479
nofonts	27
noindent	26
nopunct	27
\ <code>\normalem</code>	1701
\ <code>\normalfont</code>	540, 541, 542, 1185, 1191, 1196, 1220, 1221, 1222, 1244, 2186, 2187, 2212, 2213, 2225, 2226, 2237, 2238, 2278, 2279, 2302, 2303, 2389
\ <code>\normalsize</code>	2596, 2609, 2615, 2841, 2880, 2918, 2926
nospace	27
nozhmap	27
\ <code>\numberline</code>	83, 2251, 2252, 2334, 2335, 2348, 2349, 2432, 2433, 2439, 2440
O	
\ <code>\O</code>	106
one commands:	
\ <code>\c_one</code>	235, 1412, 1517, 1986, 2137, 3794
or commands:	
\ <code>\or:</code>	459, 2836
P	
\ <code>\P</code>	106, 197
\ <code>\pagestyle</code>	2722
pagestyle	78
\ <code>\par</code>	2165, 2189, 2194, 2215, 2218, 2219, 2227, 2228, 2239, 2240, 2282, 2287, 2288, 2305, 2306, 2518, 2533, 2553
paragraph/aftername	18
paragraph/afterskip	20
paragraph/beforeskip	19
paragraph/format	17
paragraph/indent	20
paragraph/name	15
paragraph/nameformat	17
paragraph/number	16
paragraph/numberformat	18
paragraph/titleformat	18
\ <code>\parindent</code>	1953, 1954, 1976, 1997, 1998, 2003, 2004, 2186, 2187, 2225, 2226, 2278, 2279, 2302, 2303, 2660
\ <code>\parsep</code>	2856, 2857, 2867, 2868, 2895, 2896, 2906, 2907
\ <code>\part</code>	78, 2163
part/aftername	18
part/afterskip	20
part/beforeskip	19
part/format	17
part/indent	20
part/name	15
part/nameformat	17
part/number	16
part/numberformat	18
part/pagestyle	20
part/titleformat	18
\ <code>\partname</code>	2189, 2215, 2510
\ <code>\PassOptionsToClass</code>	464, 469, 474
\ <code>\PassOptionsToPackage</code>	3, 512, 1672, 2006
\ <code>\patchcmd</code>	154
\ <code>\pdfmapline</code>	3497, 3498, 3499, 3500, 3501, 3502, 3503, 3505, 3506, 3507, 3508, 3509, 3510, 3511, 3513, 3514, 3515, 3516, 3517, 3518, 3520, 3521, 3522, 3523, 3524, 3525, 3626, 3627, 3628, 3629, 3630, 3631, 3632, 3634, 3635, 3636, 3637, 3638, 3639, 3640, 3642, 3643, 3644, 3645, 3646, 3647, 3649, 3650, 3651, 3652, 3653, 3654, 3693, 3694, 3695, 3696, 3697, 3698, 3700, 3701, 3702, 3703, 3704, 3705, 3707, 3708, 3709, 3710, 3711, 3713, 3714, 3715, 3716, 3717
\ <code>\pdfstringdefDisableCommands</code>	2931
pdftex commands:	
\ <code>\pdftex_if_engine:TF</code>	206, 1858, 3139, 3219, 3274, 3340, 3353, 3400
\ <code>\pretocmd</code>	155
prg commands:	
\ <code>\prg_do_nothing:</code>	1358, 1450, 1732, 1795, 1829
\ <code>\prg_new_conditional:Npnn</code>	40, 842, 1773
\ <code>\prg_new_protected_conditional:Npnn</code>	1048
\ <code>\prg_return_false:</code>	43, 845, 1058, 1777, 1781
\ <code>\prg_return_true:</code>	45, 845, 1051, 1056, 1777, 1781
\ <code>\primitive</code>	3850
\ <code>\ProcessKeysOptions</code>	453
prop commands:	
\ <code>\prop_get:NnN</code>	1143
\ <code>\prop_get:NnNT</code>	1351
\ <code>\prop_get:NnNTF</code>	1061, 1102, 1140, 1399, 2746, 2801, 2828
\ <code>\prop_get:NxNTF</code>	1050
\ <code>\prop_gpop:NnNT</code>	990, 994, 1486
\ <code>\prop_gput:Nnn</code>	967, 1130, 2763
\ <code>\prop_gput:Nno</code>	968, 1016, 1474, 1510
\ <code>\prop_gremove:Nn</code>	999
\ <code>\prop_if_empty:NF</code>	1064, 1225
\ <code>\prop_map_break:n</code>	1239
\ <code>\prop_map_inline:Nn</code>	1237
\ <code>\prop_new:N</code>	985, 986, 987, 1135, 1494, 1515, 2757
\ <code>\protect</code> 130, 131, 2251, 2252, 2260, 2261, 2334, 2335, 2348, 2349	
\ <code>\ProvidesFile</code>	3485, 3493, 3560, 3591, 3622, 3689
punct	10, 27, 37, 71
\ <code>\punctstyle</code>	584, 1850

- Q**
- `\quad` 2313, 2453, 2455, 2462, 2464,
2468, 2469, 2471, 2525, 2560, 2569, 2584, 2597, 2610, 2616
- `\quotation` 94, 2946
- R**
- `\raggedright` 2186, 2225, 2278, 2302, 2517, 2552
- `\refname` 2047, 2053, 2059, 2060
- `\refstepcounter` 84, 2179, 2204, 2247, 2319, 2496, 2497
- `\relax` 891, 912, 915, 916,
2167, 2203, 2214, 2271, 2295, 2320, 2322, 2326, 2342,
2357, 2367, 2368, 2462, 2464, 3449, 3453, 3456, 3482, 3484
- `\renewcommand` 2163
- `\RenewDocumentCommand` 2497
- `\RequirePackage` .. 7, 23, 410, 416, 513, 516, 517, 518, 665,
684, 687, 721, 729, 760, 1682, 1691, 1701, 1702, 2007, 2737
- `\RequirePackageWithOptions` 4
- `resetalternatefont` 24, 61
- reverse commands:
- `\reverse_if:N` 586
- `\rmdefault` 1596
- `\rmfamily` 534, 535, 1214, 1215
- S**
- `\S` 106
- scan commands:
- `\scan_stop:` ... 61, 194, 816, 1451, 1452, 1924, 1932, 3778
- scheme 10, 38
- scriptfont@name commands:
- `\scriptfont@name:` 902
- `\scriptsize` 2871, 2910
- `\secdef` 2174
- `section/aftername` 18
- `section/afterskip` 20
- `section/beforeskip` 19
- `section/format` 17
- `section/indent` 20
- `section/name` 15
- `section/nameformat` 17
- `section/number` 16
- `section/numberformat` 18
- `section/titleformat` 18
- `\selectfont` .. 1042, 1477, 1490, 1800, 1805, 1821, 1888, 2747
- seq commands:
- `\seq_clear:N` 961
- `\seq_concat:NNN` 1295
- `\seq_gput_right:Nn` 1068, 2769
- `\seq_gset_from_clist:Nn` 2070, 3758
- `\seq_if_empty:NF` 1307
- `\seq_if_in:NnF` 1066
- `\seq_map_inline:Nn` 1315, 2147, 2485, 3781
- `\seq_new:N` 1073, 1300, 1301, 2069, 2758, 3757
- `\seq_put_right:No` 1563
- `\seq_set_filter:NNn` 1293
- `\seq_set_split:NnV` 1292
- `\seq_use:Nnnn` 2755
- `\setCJKfamilyfont` 55, 1087, 1161,
3154, 3159, 3166, 3167, 3168, 3169, 3170, 3171, 3196,
3197, 3205, 3206, 3208, 3209, 3210, 3211, 3213, 3234,
3236, 3238, 3240, 3267, 3268, 3269, 3270, 3293, 3298,
3303, 3305, 3331, 3333, 3335, 3336, 3346, 3347, 3348,
3349, 3361, 3362, 3363, 3364, 3365, 3366, 3391, 3392,
3393, 3394, 3395, 3396, 3408, 3409, 3410, 3431, 3432, 3433
- `\setCJKmainfont` .. 56, 1083, 1182, 1187, 1203, 3144, 3192,
3200, 3226, 3260, 3281, 3326, 3343, 3357, 3386, 3404, 3428
- `\setCJKmathfont` 56, 1198, 1206
- `\setCJKmonofont` 56, 1085, 1193, 1205, 3165, 3195,
3204, 3233, 3266, 3292, 3330, 3345, 3360, 3390, 3407, 3430
- `\setCJKromanfont` 1187, 1207
- `\setCJKsansfont`
..... 56, 1084, 1188, 1204, 3147, 3153, 3158, 3194,
3202, 3232, 3265, 3287, 3329, 3344, 3359, 3388, 3406, 3429
- `\SetSymbolFont` 944, 1265, 1269
- `\sfdefault` 1597
- `\sffamily` 536, 537, 1216, 1217
- six commands:
- `\c_six` 2766
- skip commands:
- `\skip_horizontal:N` 1763, 1767
- `\skip_if_eq:nnT` 1567
- `\skip_if_eq:nnTF` 1780
- `\skip_new:N` 1574, 1772
- `\skip_set:Nn` ... 1570, 1575, 1756, 1925, 1936, 1981, 1989
- `\skip_zero:N` 1945
- `\small` 2849, 2888
- `\songti` 3437
- `\space` 2248, 2282, 2510, 2550, 2667
- space 11, 27, 37, 70
- `\special` 1689, 3528, 3529, 3530, 3531, 3532, 3533,
3534, 3536, 3537, 3538, 3539, 3540, 3541, 3542, 3544,
3545, 3546, 3547, 3548, 3549, 3551, 3552, 3553, 3554,
3555, 3556, 3567, 3568, 3569, 3570, 3571, 3573, 3574,
3575, 3576, 3577, 3579, 3580, 3581, 3582, 3584, 3585,
3586, 3587, 3598, 3599, 3600, 3601, 3602, 3604, 3605,
3606, 3607, 3608, 3610, 3611, 3612, 3613, 3615, 3616,
3617, 3618, 3657, 3658, 3659, 3660, 3661, 3662, 3663,
3665, 3666, 3667, 3668, 3669, 3670, 3671, 3673, 3674,
3675, 3676, 3677, 3678, 3680, 3681, 3682, 3683, 3684,
3685, 3720, 3721, 3722, 3723, 3724, 3726, 3727, 3728,
3729, 3730, 3732, 3733, 3734, 3735, 3737, 3738, 3739, 3740
- `\SplitArgument` 1412, 1517, 2137
- stop commands:
- `\q_stop` ... 137, 142, 151, 194, 202, 1316, 1320, 1547, 1552
- str commands:
- `\str_case:nnTF` 128
- `\str_case:onF` 1594
- `\str_case:onTF` 142
- `\str_case_x:nnF` 1081, 1092
- `\str_if_eq:nnF` 3027
- `\str_if_eq:nnTF` 923
- `\str_if_eq:onF` 587, 3034
- `\str_if_eq:onTF` 515, 1678, 1686, 2723, 3010

<code>\str_if_eq_x:nnF</code>	583
<code>\str_if_eq_x:nnTF</code>	134, 1229, 3007
<code>\string</code>	894, 896, 898, 907, 915
<code>\strutbox</code>	2924
<code>sub3section</code>	10, 38
<code>sub4section</code>	10, 38
<code>subparagraph/aftername</code>	18
<code>subparagraph/afterskip</code>	20
<code>subparagraph/beforeskip</code>	19
<code>subparagraph/format</code>	17
<code>subparagraph/indent</code>	20
<code>subparagraph/name</code>	15
<code>subparagraph/nameformat</code>	17
<code>subparagraph/number</code>	16
<code>subparagraph/numberformat</code>	18
<code>subparagraph/titleformat</code>	18
<code>subsection/aftername</code>	18
<code>subsection/afterskip</code>	20
<code>subsection/beforeskip</code>	19
<code>subsection/format</code>	17
<code>subsection/indent</code>	20
<code>subsection/name</code>	15
<code>subsection/nameformat</code>	17
<code>subsection/number</code>	16
<code>subsection/numberformat</code>	18
<code>subsection/titleformat</code>	18
<code>subsubsection/aftername</code>	18
<code>subsubsection/afterskip</code>	20
<code>subsubsection/beforeskip</code>	19
<code>subsubsection/format</code>	17
<code>subsubsection/indent</code>	20
<code>subsubsection/name</code>	15
<code>subsubsection/nameformat</code>	17
<code>subsubsection/number</code>	16
<code>subsubsection/numberformat</code>	18
<code>subsubsection/titleformat</code>	18
T	
<code>\T</code>	198
<code>\t</code>	105
<code>\tablename</code>	2043
<code>tablename</code>	13
<code>\TeX</code>	2928
TeX and L^AT_EX 2_ε commands:	
<code>\@@italiccorr</code>	918
<code>\@@par</code>	2327, 2329, 2362, 2364
<code>\@afterheading</code>	2198, 2232, 2266
<code>\@afterindentfalse</code>	2169, 2273, 2297
<code>\@afterindenttrue</code>	2171, 2275, 2299
<code>\@Alph</code>	2663, 2668
<code>\@chapapp</code>	2248, 2282, 2453, 2468, 2469
<code>\@chapter</code>	79, 2244
<code>\@currentlabel</code>	2476, 2480
<code>\@currentx</code>	478
<code>\@currname</code>	67, 478
<code>\@defaultunits</code>	33
<code>\@empty</code>	84, 2317
<code>\@endpart</code>	2220, 2241
<code>\@EverySelectfont@Init</code>	1802
<code>\@changefrom</code>	2326, 2361
<code>\@ifclassloaded</code>	2683, 2736
<code>\@ifpackagelater</code>	17, 19, 672, 685, 688
<code>\@ifpackageloaded</code>	50, 491, 681, 1665, 2434, 2493
<code>\@ifundefined</code>	2312, 2335, 2349
<code>\@listI</code>	2847, 2886
<code>\@listi</code>	2847, 2854, 2865, 2886, 2893, 2904
<code>\@M</code>	2185, 2211, 2224, 2236, 2286, 2304, 2327, 2328, 2362, 2363
<code>\@makechapterhead</code>	80, 2263, 2265, 2268
<code>\@makeschapterhead</code>	80, 2292
<code>\@minus</code>	2573, 2577, 2587, 2590, 2600, 2603, 2623, 2626, 2634, 2643, 2646, 2654, 2843, 2845, 2851, 2853, 2855, 2856, 2862, 2864, 2866, 2867, 2882, 2884, 2890, 2892, 2894, 2895, 2901, 2903, 2905, 2906
<code>\@nil</code>	869
<code>\@notprerr</code>	30, 30, 42
<code>\@onlypreamble</code>	30, 30, 30, 42, 528, 1203, 1204, 1205, 1206, 1207, 1208, 2994
<code>\@part</code>	78, 2174, 2177, 2201
<code>\@pkgextension</code>	480, 481, 482, 493, 683
<code>\@plus</code>	2570, 2573, 2577, 2585, 2587, 2590, 2598, 2600, 2603, 2623, 2626, 2628, 2634, 2643, 2646, 2648, 2654, 2843, 2844, 2845, 2851, 2852, 2853, 2855, 2856, 2862, 2863, 2864, 2866, 2867, 2882, 2883, 2884, 2890, 2891, 2892, 2894, 2895, 2901, 2902, 2903, 2905, 2906
<code>\@popfilename</code>	30, 66
<code>\@secCNTformat</code>	81, 2311, 2320
<code>\@sect</code>	81, 2315
<code>\@setfontsize</code>	2833
<code>\@spart</code>	79, 2174, 2223, 2235
<code>\@ssect</code>	82, 82, 2356
<code>\@startsection</code>	82, 82, 85, 2382
<code>\@svsec</code>	2317, 2320, 2326, 2343, 2344
<code>\@svsechd</code>	2341, 2367, 2368
<code>\@tempdima</code>	2420, 2422
<code>\@tempskipa</code>	2167, 2168, 2169, 2173, 2271, 2272, 2273, 2277, 2295, 2296, 2297, 2301, 2322, 2323, 2357, 2358
<code>\@topnewpage</code>	2263
<code>\@xsect</code>	2355, 2370
<code>\abstractname</code>	13
<code>\addCJKfontfeature</code>	61
<code>\AfterEndEnvironment</code>	94
<code>\appendixname</code>	13, 21, 21, 21
<code>\AtBeginDocument</code>	42
<code>\AtEndEnvironment</code>	94
<code>\AtEndOfClass</code>	34
<code>\AtEndOfPackage</code>	34
<code>\AtEndOfPackageFile*</code>	30
<code>\baselineskip</code>	11, 12, 64, 72, 72
<code>\bfseries</code>	44
<code>\bibname</code>	13
<code>\c@chapter</code>	2668

<code>\c@secnumdepth</code>	2178, 2188, 2203, 2214, 2245, 2280, 2316, 2333, 2347
<code>\c@section</code>	2663
<code>\caption</code>	28
<code>\catcode</code>	30, 30, 41, 41, 104
<code>\ccwd</code> ..	12, 12, 12, 22, 22, 22, 27, 33, 68, 69, 70, 72, 72, 72, 73
<code>\chapter</code>	10, 15, 20, 21, 21, 79
<code>\chaptermark</code>	14
<code>\chaptername</code>	16
<code>\char</code>	46
<code>\CheckCommand</code>	70
<code>\Chinese</code>	28, 28
<code>\chinese</code>	22, 22, 28, 28
<code>\CJK@@ignorespaces</code>	43, 600
<code>\CJK@encoding</code>	51, 51, 52, 60, 775, 779, 863, 930, 931, 933, 935, 936, 937, 939, 943, 945, 1260, 1263, 1265, 1269, 1409, 1424, 1425, 1442, 1443
<code>\CJK@envStart</code>	42, 42, 42, 594
<code>\CJK@family</code>	47, 47, 770, 772, 775, 780, 827, 863, 1039, 1476, 1486, 1488, 1821, 1822
<code>\CJK@ignorespaces</code>	43, 43, 605, 609
<code>\CJK@input</code>	41, 42, 559
<code>\CJK@loadBinding</code>	41, 42, 590
<code>\CJK@loadEncoding</code>	41
<code>\CJK@makeActive</code>	42, 589
<code>\CJK@plane</code>	41, 70, 1809
<code>\CJK@surr</code>	41, 41, 41, 560, 579
<code>\CJK@upperReset</code>	42
<code>\CJKfamily</code>	52
<code>\CJKfamilydefault</code>	56, 64, 64, 64, 65, 65
<code>\CJKglue</code>	72, 72, 93
<code>\CJKhook</code>	43, 43
<code>\CJKkrmdefault</code>	41
<code>\CJKsymbol</code>	70
<code>\CJKunderdotbasesep</code>	27
<code>\CJKunderline</code>	27
<code>\clearpage</code>	43
<code>\contentsname</code>	13, 13
<code>\CS</code>	48, 48
<code>\csname</code>	93, 94
<code>\CTEX@appendixnumber</code>	2400, 2408, 2413
<code>\CTEX@chapter@afterskip</code>	2290, 2308
<code>\CTEX@chapter@beforeskip</code>	2271, 2295
<code>\CTEX@chapter@format</code>	2279, 2303
<code>\CTEX@chapter@indent</code>	2279, 2303
<code>\CTEX@chapter@pagestyle</code>	2158, 2269, 2293
<code>\CTEX@chapter@titleformat</code>	2288, 2306
<code>\CTEX@chaptername</code>	2283
<code>\CTEX@part@afterskip</code>	2197, 2231
<code>\CTEX@part@beforeskip</code>	2167
<code>\CTEX@part@format</code>	2187, 2213, 2226, 2238
<code>\CTEX@part@indent</code>	2187, 2226
<code>\CTEX@part@pagestyle</code>	2157, 2202
<code>\CTEX@part@titleformat</code>	2193, 2219, 2228, 2240
<code>\CTEX@partname</code>	2190, 2216
<code>\CTEX@postappendix</code>	2394, 2409, 2414
<code>\CTEX@postchapter</code>	2414
<code>\CTEX@postsection</code>	2409
<code>\CTEX@preappendix</code>	2393, 2407, 2412
<code>\CTEX@prechapter</code>	2412
<code>\CTEX@presection</code>	2407
<code>\CTEX@save@appendix</code>	2402, 2405
<code>\CTEX@save@refstepcounter</code>	2496, 2499
<code>\CTEX@selectfont@hook</code>	70, 1792, 1793, 1795, 1800, 1809, 1812
<code>\CTEX@set@titleformat@n</code>	82, 2372, 2388
<code>\CTEX@setcurrentlabel@n</code>	2474, 2500
<code>\CTEX@thechapter</code>	2413
<code>\CTEX@thesection</code>	2408
<code>\CTEX@titleformat@n</code> ..	82, 82, 82, 2364, 2368, 2373, 2374
<code>\CTEX@toc@width@n</code>	2417, 2432, 2439
<code>\CTEX@todayold</code>	2015, 2020
<code>\CTEX@verbatim@font@hook</code>	1578, 1581
<code>\ctex@zhmap@endinput</code>	3745
<code>\CTEXdigits</code>	23, 23
<code>\CTEXnumber</code>	23, 23, 23, 23, 23
<code>\CTEXoptions</code>	28, 28
<code>\ctexset</code>	6, 6, 6, 6, 6, 6, 6, 8, 8, 10, 11, 12, 12, 12, 12, 13, 13, 13, 13, 13, 13, 13, 13, 13, 15, 15, 16, 16, 17, 17, 18, 18, 18, 19, 20, 20, 20, 21, 21, 24, 24, 26, 27, 28, 95
<code>\CTEXsetup</code>	27
<code>\CTEXthechapter</code>	14, 14, 14, 16
<code>\CTEXthesection</code>	14, 14, 16, 84
<code>\CTEXunderdotbasesep</code>	27
<code>\CTEXunderline</code>	27
<code>\curr@fontshape</code>	781, 811, 820, 824, 870
<code>\DeclareAlternateKanjiFont</code>	60
<code>\DeclareFontFamily</code>	48
<code>\DeclareFontShape</code>	48
<code>\DeclareFontShape@</code>	60, 1368
<code>\DeclareRobustCommand</code>	31
<code>\define@newfont</code>	47, 803
<code>\do@subst@correction</code>	47, 48, 802
<code>\end</code>	94, 94
<code>\EverySelectfont</code>	70
<code>\external@font</code>	813, 814, 816
<code>\extract@font</code>	47, 801
<code>\f@baselineskip</code>	1924
<code>\f@encoding</code>	779, 819, 914
<code>\f@family</code>	47, 780, 819, 827
<code>\f@series</code>	775, 863
<code>\f@shape</code>	775, 863
<code>\f@size</code>	775, 781, 858, 866, 869, 870, 1752, 1788, 2423
<code>\false</code>	9
<code>\fangsong</code>	9, 9
<code>\figurename</code>	13
<code>\font</code>	48, 48
<code>\font@name</code>	47, 48, 784, 791, 792, 795, 798, 816, 817, 834, 894, 895, 896, 897, 898
<code>\fontcharwd</code>	69, 69
<code>\footnotesep</code>	93, 93, 93
<code>\footnotesize</code>	93

<code>\get@external@font</code>	810	<code>\normalsize</code>	9, 9, 89
<code>\getanddefine@fonts</code>	892, 906	<code>\nouppercase</code>	41
<code>\globaljfont</code>	48	<code>\p@</code>	1752, 1788, 2215, 2260, 2261, 2270, 2282, 2289, 2294, 2307, 2423, 2533, 2548, 2553, 2555, 2562, 2843, 2844, 2845, 2851, 2852, 2853, 2855, 2856, 2862, 2863, 2864, 2866, 2867, 2882, 2883, 2884, 2890, 2891, 2892, 2894, 2895, 2901, 2902, 2903, 2905, 2906
<code>\heiti</code>	9	<code>\pagestyle</code>	10
<code>\Hy@driver</code>	1686	<code>\paragraph</code>	10, 10, 10, 10, 10, 19, 20, 33, 33
<code>\Hy@unicodetrue</code>	67	<code>\parindent</code>	12, 22, 72, 74, 93
<code>\HyPsd@ConvertToUnicode</code>	67	<code>\part</code>	15, 20, 79
<code>\HyPsd@LoadUnicode</code>	67	<code>\partname</code>	16
<code>\HyPsd@pdfencoding</code>	67, 67	<code>\PassOptionsToClass</code>	39
<code>\HyPsd@ToBigChars</code>	67	<code>\PassOptionsToPackage</code>	66
<code>\if@mainmatter</code>	2246, 2281, 2468, 2690, 2691	<code>\patchcmd</code>	31
<code>\if@noskipsec</code>	2164	<code>\pdfmapline</code>	99, 101
<code>\if@twocolumn</code>	2262	<code>\pdfstringdef</code>	67
<code>\if@twoside</code>	2447, 2454	<code>\pickup@font</code>	47
<code>\ifin@</code>	908, 916	<code>\pickup@jfont</code>	807, 895, 897, 899
<code>\ifpdf</code>	29	<code>\ProcessKeysOptions</code>	25
<code>\indexname</code>	13	<code>\protected</code>	46
<code>\itdefault</code>	48	<code>\protected@edef</code>	2012, 2014, 2320, 2476
<code>\itshape</code>	44	<code>\ProvidesFile</code>	104
<code>\kaishu</code>	9	<code>\ps@fancy</code>	84, 2461, 2463, 2467, 2470
<code>\labelformat</code>	84, 84	<code>\ps@headings</code>	83, 2446, 2448, 2452, 2455
<code>\leftmark</code>	14	<code>\refname</code>	13
<code>\linewidth</code>	72, 72	<code>\refstepcounter</code>	84
<code>\lishu</code>	9, 9	<code>\relax</code>	41, 72
<code>\listfigurename</code>	13	<code>\reset@font</code>	541, 1221
<code>\listtablename</code>	13	<code>\rightmark</code>	14
<code>\ltj@al@getanddefine@fonts</code>	892, 909	<code>\rmfamily</code>	45
<code>\ltj@does@alt@set</code>	49	<code>\roman</code>	22
<code>\ltj@getjfontnumber</code>	50, 879	<code>\scantokens</code>	31
<code>\ltj@IsFontJapanese</code>	51, 887, 891, 907	<code>\scriptfont@name</code>	897
<code>\ltj@ja@getanddefine@fonts</code>	893, 908	<code>\section</code>	10, 10, 15, 15, 21, 21
<code>\ltj@mathJapaneseFonts</code>	891, 915	<code>\sectionmark</code>	14, 14
<code>\ltj@set@stackfont</code>	901, 902, 903	<code>\selectfont</code>	49, 64, 70, 70, 70, 72
<code>\ltj@allalchar</code>	726, 744	<code>\setCJKfamilyfont</code>	23, 23, 24
<code>\ltj@curjfnt</code>	48, 70	<code>\setCJKmainfont</code>	24
<code>\ltj@pickup@altfont@auxy</code>	49, 864	<code>\sf@size</code>	896
<code>\ltj@pickup@altfont@copy</code>	50, 877	<code>\ssfamly</code>	45
<code>\ltj@setpar@global</code>	900	<code>\size@update</code>	72, 72, 1888, 1902
<code>\ltj@temp</code>	908, 909, 910	<code>\sldefault</code>	48
<code>\ltj@tempcnta</code>	907, 915	<code>\songti</code>	8
<code>\ltj@tempcntc</code>	50, 883	<code>\special</code>	8, 66
<code>\ltjalchar</code>	46	<code>\split@name</code>	869
<code>\ltjsetparameter</code>	64	<code>\SplitArgument</code>	77
<code>\m@ne</code>	2178, 2188, 2245, 2280, 3054, 3055, 3056, 3057, 3058, 3059	<code>\ssf@size</code>	898
<code>\markboth</code>	14	<code>\strutbox</code>	93
<code>\markright</code>	14	<code>\subparagraph</code>	10, 10, 10, 10, 19, 20, 33
<code>\math@bgroup</code>	913	<code>\subst@correction</code>	830, 837
<code>\math@egroup</code>	917	<code>\tablename</code>	13
<code>\math@fonts</code>	900	<code>\textfont@name</code>	895
<code>\mathbb</code>	93	<code>\tf@size</code>	894
<code>\maxdimen</code>	12, 72	<code>\thechapter</code>	14, 16, 21
<code>\meaning</code>	31	<code>\thesection</code>	14, 16, 16, 21, 84
<code>\newcommand</code>	31		
<code>\newrobustcmd</code>	31		
<code>\nfss@catcodes</code>	51		

<code>\thispagestyle</code>	79	<code>\tl_if_blank:nTF</code>	1325, 1421, 1422, 1526, 1527, 1541, 1559
<code>\today</code>	12, 12	<code>\tl_if_blank_p:n</code>	1294
<code>\ttfamily</code>	45	<code>\tl_if_empty:cT</code>	2490
<code>\uppercase</code>	41, 41	<code>\tl_if_empty:NF</code>	770, 1951
<code>\use@mathgroup</code>	912	<code>\tl_if_empty:NTF</code>	2090, 2975
<code>\usepackage</code>	27, 27	<code>\tl_if_eq:NNT</code>	1588
<code>\verbatim@font</code>	1578, 1579	<code>\tl_if_eq:NNTF</code>	1891
<code>\wrong@fontshape</code>	47	<code>\tl_if_exist:NF</code>	
<code>\x@protect</code>	131	56, 529, 531, 532, 533, 1209, 1211, 1212, 1213, 2686
<code>\XeTeXglyphbounds</code>	109	<code>\tl_if_exist:NTF</code>	90, 2056, 2059
<code>\yahei</code>	9, 9	<code>\tl_map_function:nN</code>	106
<code>\youyuan</code>	9, 9	<code>\tl_map_inline:Nn</code>	1437
<code>\z@</code>	713, 716, 2168, 2186, 2225, 2272, 2278, 2296, 2302, 2323, 2358, 2844, 2852, 2863, 2883, 2891, 2902	<code>\tl_map_inline:nn</code>	2799
<code>\zhdigits</code>	23	<code>\tl_new:c</code>	2078, 2079
<code>\zhnumber</code>	22, 23, 23, 23	<code>\tl_new:N</code>	24, 56, 204, 209, 531, 532, 533, 772, 773, 861, 927, 978, 1004, 1024, 1025, 1046, 1373, 1514, 1609, 1899, 1912, 2393, 2394
<code>\zihao</code>	21, 21	<code>\tl_put_left:Nn</code>	66, 1802
<code>\ziju</code>	22	<code>\tl_put_right:Nn</code>	428, 1554, 2119, 2121, 2123, 2125, 2127
<code>\zw</code>	64, 64, 69	<code>\tl_put_right:Nx</code>	1521, 2104, 2802, 2803
tex commands:		<code>\tl_replace_all:Nnn</code>	1291, 1329
<code>\tex_divide:D</code>	72, 1933, 1974	<code>\tl_set:cn</code>	2141, 2144
<code>\tex_endlinechar:D</code>	550, 556	<code>\tl_set:Nn</code>	188, 189, 261, 294, 295, 521, 522, 523, 761, 763, 774, 862, 924, 925, 962, 1029, 1041, 1290, 1326, 1328, 1369, 1542, 1546, 1821, 1864, 1875, 1913, 2932, 3185, 3186, 3187
<code>\tex_font:D</code>	3778	<code>\tl_set:Nx</code> ...	205, 563, 813, 1165, 1611, 1849, 1860, 1961
<code>\tex_iftrue:D</code>	2691	<code>\tl_set_eq:Nc</code>	478
<code>\tex_ignorespaces:D</code>	1036, 1174, 1179, 1835, 1958, 2743	<code>\tl_set_eq:NN</code>	
<code>\tex_lowercase:D</code>	107, 199, 736, 1283	779, 780, 827, 935, 1030, 1055, 1476, 1488, 1900
<code>\tex_uppercase:D</code>	1651	<code>\tl_trim_spaces:n</code>	1505
textfont@name commands:		<code>\today</code>	2015, 2020, 2023, 2028
<code>\textfont@name:</code>	901	<code>today</code>	12, <u>75</u>
<code>\the</code>	883, 3453, 3456, 3457, 3476, 3480	token commands:	
<code>\thechapter</code>	2248, 2251, 2282, 2453, 2468, 2469, 2551	<code>\token_get_replacement_spec:N</code>	117, 137
<code>\theparagraph</code>	2608	<code>\token_to_str:N</code>	178, 1083, 1084, 1085, 1087, 1094, 1095, 1096, 1158, 2053, 2054, 2751, 3785
<code>\thepart</code>	2180, 2189, 2205, 2215, 2511	<code>\toks</code>	3453, 3454, 3456, 3457, 3476, 3480
<code>\thesection</code>	2446, 2455, 2462, 2471, 2568	<code>\topsep</code>	2855, 2866, 2894, 2905
<code>\thesubparagraph</code>	2614	<code>\TrimSpaces</code>	497
<code>\thesubsection</code>	2448, 2464, 2582	<code>\ttdefault</code>	1598
<code>\thesubsubsection</code>	2595	<code>\ttfamily</code>	538, 539, 1218, 1219
<code>\thispagestyle</code>	2202, 2269, 2293	two commands:	
three commands:		<code>\c_two</code>	207, 236, 432, 436, 1979, 2423, 2618, 2658
<code>\c_three</code>	396, 1978, 2638, 3795	<code>\c_two_hundred_fifty_six</code>	566, 569
<code>\tiny</code>	2872, 2911	<code>\typeout</code>	2248, 2249
tl commands:			
<code>\tl_clear:c</code>	1712, 2143	U	
<code>\tl_clear:N</code>		um commands:	
..	266, 421, 444, 1508, 1512, 1822, 1879, 2146, 2153, 2798	<code>\um_cs_set_eq_active_char:Nw</code>	753, 754
<code>\tl_const:cn</code>	660, 683	<code>\undefined</code>	3478
<code>\tl_const:cx</code>	2080, 2086	<code>\updefault</code> ...	932, 937, 939, 943, 945, 1261, 1263, 1266, 1270
<code>\tl_const:Nn</code>	463, 468, 473, 530, 930, 941, 1210, 1211, 1212, 1213, 1653, 1662, 2684, 2692, 2694	use commands:	
<code>\tl_const:Nx</code>	1259, 1898	<code>\use:c</code>	781, 819, 820, 870, 946, 996, 1489
<code>\tl_gput_right:Nn</code>	57	<code>\use:n</code>	210, 1230, 1234, 1610, 1730, 1797, 2374, 2428, 2951
<code>\tl_gset:Nn</code>	1233, 1649, 1659, 1660		
<code>\tl_gset:Nx</code>	567, 570, 1592, 1621, 3012		
<code>\tl_gset_eq:NN</code>	1018, 1616		
<code>\tl_gset_rescan:Nnn</code>	1240		
<code>\tl_head:N</code>	2802		

<code>\use:x</code>	70,	xetex commands:	
	115, 443, 758, 869, 977, 1113, 1331, 1505, 2152, 2761, 2984	<code>\xetex_if_engine:F</code>	3751
<code>\use_i:nn</code>	36	<code>\xetex_if_engine:TF</code>	1865, 3389
<code>\use_i:nnn</code>	334	<code>\XeTeXcharglyph</code>	3782
<code>\use_ii:nn</code>	38	<code>\XeTeXglyphbounds</code>	3806
<code>\use_ii:nnn</code>	211, 339		
<code>\use_iii:nnn</code>	344		
<code>\use_none:n</code> ..	262, 267, 272, 1013, 1233, 1366, 1731, 2010		
UTF8	7, 36		
	V		Y
<code>\verse</code>	94, 2944, 2945	<code>\yahei</code>	3445
<code>\vskip</code>	2196, 2197,	<code>\youyuan</code>	3443
	2215, 2230, 2231, 2282, 2289, 2290, 2307, 2308, 2533, 2553		
<code>\vspace</code>	2270, 2277, 2294, 2301		
	W		Z
<code>winfonts</code>	27	zero commands:	
<code>\write</code>	3487	<code>\c_zero</code>	234, 731, 735, 1934, 2728
		<code>\c_zero_dim</code>	1926, 1938, 1953, 1969, 1976, 2130, 2659
		<code>\c_zero_fp</code>	1898, 2729, 2920
		<code>\c_zero_skip</code>	2128, 2129
		<code>\zhdigits</code>	23, 2014
		<code>zhmap</code>	8, 27, 37
		<code>\zhnum</code>	2008
		<code>\zhnumber</code>	22, 2012
		<code>\zhnumsetup</code>	2024, 2029
		<code>\zhtoday</code>	2023, 2028
		<code>\zihao</code>	21, 89, 2742, 2751
		<code>zihao</code>	9, 35
		<code>\ziju</code>	22, 73, 1957
		<code>\zw</code>	714, 1570, 1575, 1756, 1789
<code>\x</code>	3455, 3458,		
	3459, 3460, 3461, 3462, 3463, 3464, 3465, 3466, 3467,		
	3468, 3469, 3470, 3471, 3472, 3473, 3474, 3481, 3485, 3489		
<code>\xdef</code>	894, 896, 898, 3488		
<code>\xeCJKsetup</code>	666, 1767, 1830, 1833, 1836, 1851		