

CT_EX 宏集手册

CTEX.ORG

2021/03/14 v2.5.6*

简介

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

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX、upL^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` 以及中文文档类 `ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer` 实现。

目录

第 1 节 介绍	2	第 7 节 章节标题样式设置	15
第 2 节 简明教程	3	7.1 编号相关	16
2.1 CT _E X 宏集的组成	3	7.2 格式相关	17
2.2 CT _E X 宏集的安装和更新	3	7.3 间距、缩进相关	22
2.3 使用 CT _E X 文档类	4	7.4 目录、附录相关	25
2.4 使用 <code>ctex</code> 宏包	5	7.5 辅助命令	26
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	7.6 示例	27
第 4 节 编译方式、编码与中文字库	6	第 8 节 实用命令	27
4.1 编译方式	6	8.1 字号与间距	27
4.2 中文编码	6	8.2 中文数字转换	27
4.3 中文字库	6	8.3 杂项	29
第 5 节 排版格式设定	8	第 9 节 LuaL ^A T _E X 下的中文支持方式	29
5.1 文档默认字号	8	9.1 LuaL ^A T _E X 下替代字体的设置	29
5.2 章节标题风格	9	第 10 节 CT _E X 宏集的配置文	30
5.3 排版方案选项	10	10.1 修改宏包默认选项	31
第 6 节 文档汉化	11	10.2 宏包载入后的配置	31
6.1 日期汉化	11	10.3 配置标题中文翻译	31
6.2 文档标题汉化	12	10.4 自定义字体集	32
6.3 页面格式设置与汉化	13	第 11 节 对旧版本的兼容性	32
		11.1 CT _E X 0.8a 及以前的版本	32
		11.2 CT _E X 0.9–CT _E X 1.0d	32
		11.3 CT _E X 1.02c 以后的 SVN 开发版	34
		11.4 CT _E X 2.2 之前的版本	34
		11.5 CT _E X 2.4.1 和 2.4.2	34

*`ctex-kit` rev. 07425fe.

11.6 CTeX 2.5 之前的版本	34	第 14 节 代码实现	38
第 12 节 宏集依赖情况与手工安装方法	35	版本历史	147
第 13 节 开发人员	36	代码索引	149

第 1 节 介绍

历史

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

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

2009 年 5 月，我们在 Google Code 建立了 `ctex-kit` 项目¹，对 `ctex` 宏包及相关脚本进行了整合，并加入了对 XeTeX 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^AT_EX 的最新发展，特别是 L^AT_EX3 的逐渐成熟，李清用 L^AT_EX3 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 CTeX 宏集。

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文字支持，特别是对多字节的中日韩表意文字的支持。这使得 T_EX 以至后来的 L^AT_EX 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^AT_EX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文书写习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 CTeX 宏集的目的之一就是解决这些 L^AT_EX 文档的汉化难题。

另一方面，随着 T_EX 引擎和 L^AT_EX 宏包的不断发展，L^AT_EX 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 CTeX 宏集的另一个主要目的就是尽可能消除这些差异带来的影响，使用户能够以一个统一的接口来使用不同的中文支持方式，使得同一份文档能够在不同环境下交换使用。

CTeX 宏集的许多实现细节离不开热心朋友们在 `bbs.ctex.org` 论坛⁴上的讨论，在此对参与讨论的朋友们表示感谢。

关于宏集名字的说明

CTeX 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“CTeX”。

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

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

³比如：pdfTeX 引擎下的 CJK、zhmCJK 宏包，XeTeX 引擎下的 xeCJK 宏包和 LuaTeX 引擎下的 LuaTeX-jan 宏包。

⁴2018 年，CTeX 论坛因故无限期关闭，此链接现已失效。

CT_EX 宏集是由 CT_EX 社区发起并维护的 L^AT_EX 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 宏集并非是一事物。

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

第 2 节 简明教程

2.1 CT_EX 宏集的组成

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

表 1 CT_EX 宏集的组成

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

2.2 CT_EX 宏集的安装和更新

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

T_EX Live 的包管理器是 tlmgr(T_EX Live Manager)。用户可以在系统命令行中⁶执行

```
tlmgr gui
```

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

```
tlmgr install ctex
```

来安装 CT_EX 宏集⁷。

⁵zhmCJK 宏包是个例外。当用户显式指定选项 zhmap = zhmCJK 时,CT_EX 宏集依赖它。由于,它没有被 T_EX Live 和 MiK_T_EX 收录,用户可能需要遵照其说明文档自行安装。

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

⁷*nix 用户可能需要超级用户权限(sudo)才能正确安装宏集。

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

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

来安装 CT_EX 宏集。

若希望了解 CT_EX 宏集具体的依赖情况或手工安装宏集的方法,请参阅第 12 节。

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

对于 T_EX Live,可以在 tlmgr 的图形界面点击“更新全部已安装的”按钮或者在命令行执行

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 MiKTeX,在 MiKTeX Console 中找到“Updates”选项卡,检查更新后即可选择升级宏包。也可以使用 mpm,在命令行执行

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

来进行更新。

2.3 使用 CT_EX 文档类

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

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

例 1

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

以下是使用 ctexbeamer 文档类编写中文演示文稿的一个示例。

例 2

```
\documentclass{ctexbeamer}
\begin{document}
\begin{frame}{中文演示文稿}
\begin{itemize}
\item 你需要将所有源文件保存为 UTF-8 编码
\item 你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译
\item 也可以使用 (pdf)LaTeX 编译
\end{itemize}
\end{frame}
\end{document}
```

⁸使用 (pdf)L^AT_EX 时也能够使用 GBK 编码,但不推荐。(见 4.2 节)

```

\item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\item 对高级用户，我们也推荐使用 upLaTeX 编译
\end{itemize}
\end{frame}
\end{document}

```

2.4 使用 `ctex` 宏包

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

对于建立在 L^AT_EX 标准文档类之上开发的文档类，在使用 `ctex` 宏包时加上 `heading` 选项，可以将章节标题设置为中文风格。

例 3

```

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

```

第 3 节 宏包选项与 `\ctexset` 命令

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

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

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

后续文档将在使用说明中对某些特殊的选项加以说明。

`\ctexset`

New: 2014-03-18

`\ctexset {<键值列表>}`

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

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

例 4

```

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

```

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

第 4 节 编译方式、编码与中文字库

4.1 编译方式

$\text{CT}_{\text{E}}\text{X}$ 宏集会根据用户使用的编译方式⁹,在底层选择不同的中文支持方式(见表 2)。

表 2 $\text{CT}_{\text{E}}\text{X}$ 宏集的中文支持方式

编译方式	(pdf) \LaTeX	$\text{X}_{\text{L}}\text{\LaTeX}$	$\text{Lua}\text{\LaTeX}$	$\text{up}\text{\LaTeX}$ [*]
支持宏包	CJK	xeCJK	$\text{LuaTeX-j}\text{\LaTeX}$	原生

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

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

4.2 中文编码

GBK	☆	指明编写文档时使用的编码。 $\text{CT}_{\text{E}}\text{X}$ 宏集无法检测文档源文件的实际编码格式,因此需要用户通过选项声明。如果没有显式指定,则默认采用 UTF-8 编码。
UTF8	☆	
Updated: 2019-11-10		

使用 $\text{Xe}_{\text{L}}\text{TeX}$ 、 $\text{Lua}_{\text{L}}\text{TeX}$ 或 $\text{up}_{\text{L}}\text{TeX}$ 编译时, $\text{CT}_{\text{E}}\text{X}$ 宏集强制使用 UTF-8 编码,此时 GBK

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

用户需要保证编译方式、源文件编码、宏包编码选项三者一致。

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

4.3 中文字库

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

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

zhmap	☆	<code>zhmap = <true false zhmCJK></code>
Updated: 2014-03-08		
true		指定字体映射机制。本选项只在使用 (pdf) \LaTeX 编译时有意义。 这是该选项的默认值。 ¹¹ 使用 <code>zhmetrics</code> 宏包,将 CJK 字库通过 <code>\special</code> 命令映射到 .ttf 文件。

⁹ \LaTeX 、pdf \LaTeX 、 $\text{X}_{\text{L}}\text{\LaTeX}$ 、 $\text{Lua}\text{\LaTeX}$ 及 $\text{up}\text{\LaTeX}$ 。
¹⁰ $\text{CT}_{\text{E}}\text{X}$ 宏集现在能够识别 macOS 及 Windows 系统,并将其他系统统一归为 Linux。
¹¹ 对于以键值对形式提供的选项,文档以**粗体**来表示其默认值。下同,不再额外解释。

表 3 CT_EX 宏集自动配置字体策略

	macOS Old ¹	macOS New ²	Windows ³	其他
X _Y L ^A T _E X	xeCJK 华文字库	xeCJK 华文字库 + 苹方	xeCJK 中易字库 + 微软雅黑	xeCJK Fandol 字库 ⁴
LuaL ^A T _E X ⁵	LuaTeX-ja 华文字库	LuaTeX-ja 华文字库 + 苹方	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja Fandol 字库
pdfL ^A T _E X	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	不可用
L ^A T _E X + DVIPDFM _x	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	CJK + zhmetrics Fandol 字库
upL ^A T _E X + DVIPDFM _x	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

1 Yosemite (10.10) 及以前的 macOS 系统。
2 El Capitan (10.11) 及以后的 macOS 系统。
3 仅支持 Windows Vista 及以后的 Windows 操作系统。
4 由马起园、苏杰、黄晨成等人开发的开源中文字体,参见:<https://www.ctan.org/pkg/fandol>。
5 LuaL^AT_EX 编译时使用 LuaTeX-ja 宏包。对此,第 9 节有特别说明。
6 微软雅黑字体并不总是有效,这和选项 zhmap 的取值有关。

false 使用传统的 CJK 字库(Type 1)¹²。
zhmCJK 载入 zhmCJK 宏包¹³,由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

fontset	fontset = {adobe fandol founder mac macnew macold ubuntu windows none ...}
New: 2014-03-08	指定 CT _E X 宏集加载的字库。

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

CT_EX 预定义了以下六种中文字库。

- adobe 使用 Adobe 公司的四款中文字体,不支持 pdfL^AT_EX。
- fandol 使用 Fandol 中文字体,不支持 pdfL^AT_EX。
- founder 使用方正公司的中文字体。
- mac 使用 macOS 系统下的字体,不支持 pdfL^AT_EX,根据版本分为 macnew 和 macold 两种。
- macnew 使用 El Capitan 或之后的多字重华文字体和苹方字体。
- macold 使用 Yosemite 或之前的华文字体。
- ubuntu 使用 Ubuntu 系统下的思源宋体、思源黑体和 T_EX 发行版自带的文鼎楷体,不支持 pdfL^AT_EX。
- windows 使用 Windows 系统下的中易字体和微软雅黑字体。当使用 (pdf)L^AT_EX 编译时,微软雅黑仅在以下两种情形有效:安装有 zhmCJK 宏包且选项 zhmap=zhmCJK 时,或者安装有微软雅黑的 Type1 字体且选项 zhmap=false 时。

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

- none 不配置中文字体,需要用户自己配置。
- (name) 这里 (name) 为自定义的名字。CT_EX 宏集将载入名为 ctex-fontset-(name).def 的文件作为字体配置文件。因此,请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录下

¹²使用 (pdf)L^AT_EX 编译时,如果需要使用自定义的字体映射文件(比如需要使用 L^AT_EX + Dvips 编译),或者希望使用 Type1 字库,请禁用本选项。为此,你可能需要安装 CJK 字体。参考 zhmetrics 宏包提供的脚本 CTeXFonts.lua。
¹³zhmCJK 宏包基于 zhmetrics 和 CJK 宏包,提供与 xeCJK 宏包类似的用户接口。

合适位置建立一个名为 `ctex-fontset-<name>.def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=<name>` 选项来调用它。字体配置文件的具体写法可以参考 C_TE_X 宏集 `fontset` 目录下的字体配置文件。

注意: 如果希望使用 `\ctexset` 在导言区指定字库, 则需要先在宏包/文档类选项中指定 `fontset = none`(这会禁用 C_TE_X 宏集的操作系统检测功能和自动设定字库功能)。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 \verb|\ctexset|
指定字体。
\end{document}
```

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

`\songti` 宋体, CJK 等价命令 `\CJKfamily{zh song}`。
`\heiti` 黑体, CJK 等价命令 `\CJKfamily{zh hei}`。
`\fangsong` 仿宋, CJK 等价命令 `\CJKfamily{zh fs}`。
`\kaishu` 楷书, CJK 等价命令 `\CJKfamily{zh kai}`。

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

`\lishu` 隶书, CJK 等价命令 `\CJKfamily{zh li}`。
`\youyuan` 圆体, CJK 等价命令 `\CJKfamily{zh you}`。

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

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

在 `macnew` 字库中, 还定义了 `\pingfang`:

`\pingfang` 苹方黑体, CJK 等价命令 `\CJKfamily{zh pf}`。

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆ `zihao = <-4|5|false>`

New: 2015-05-06

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

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

10pt ☆
11pt ☆
12pt ☆

C_TE_X 文档类是在 L_AT_EX 标准文档类之上开发的。因此, 除了可以使用 C_TE_X 宏包定义的字号选项之外, 还可以使用标准文档类的同类选项 (10pt、11pt 和 12pt)。在使用这些来自标准文档类的选项的时候, C_TE_X 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。

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
\tiny	七号	5.5	小六	6.5	5	6	6
\scriptsize	小六	6.5	六号	7.5	7	8	8
\footnotesize	六号	7.5	小五	9	8	9	10
\small	小五	9	五号	10.5	9	10	11
\normalsize	五号	10.5	小四	12	10	11	12
\large	小四	12	小三	15	12	12	14
\Large	小三	15	小二	18	14	14	17
\LARGE	小二	18	二号	22	17	17	20
\huge	二号	22	小一	24	20	20	25
\Huge	一号	26	一号	26	25	25	25

标准文档类的其他选项在 CTeX 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。CTeX 会将这些选项传给标准文档类¹⁴。

5.2 章节标题风格

heading ☆

heading = (true|false)

New: 2014-03-08

本选项只能在调用 ctex.sty 时作为宏包选项使用。

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

CTeX 宏集提供的四个文档类总是启用该功能。如果在 ctex.sty 下启用该选项,将会检查当前是否使用 L^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 CTeX 宏集提供的四个中文文档类完全一致;若不然,则会根据 \chapter 是否有定义来使用 ctexbook 或者 ctexart 的标题设置。

sub3section ☆

sub4section ☆

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。

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

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

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

¹⁴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,也可能会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。

5.3 排版方案选项

scheme ☆ `scheme = <chinese|plain>`

New: 2015-04-15

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

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

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

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

punct `punct = <quanjiao|banjiao|kaiming|CCT|plain>`

Updated: 2014-04-11

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¹⁶用占一个汉字宽度, 标号和句内点号占半个汉字宽度;

CCT CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

plain 原样(不调整标点间距)。

space `space = <true|false|auto>`

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。该选项仅在使用 `XgLATEX`/(pdf)`LATEX` 编译时有效。

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

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

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

例如, 使用

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
```

¹⁵使用 `CTEX` 文档类, 或者使用 `ctex` 宏包且开启该选项时。

¹⁶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。

¹⁷`LATEX` 将单个换行视作一个空格。

则会得到“汉字分词技术 English”。

使用 `LuaATEX` 及 `upATEX` 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>`

New: 2014-04-23

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

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

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

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

New: 2014-03-13

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

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

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

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

例 8

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

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

New: 2014-03-26

`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 6 节 文档汉化

6.1 日期汉化

`CTEX` 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如本文档编译时的日期是“2021 年 3 月 14 日”。

`today` ☆ `today = <small|big|old>`

该选项用来控制 `\today` 命令的输出格式:

`small` 效果为“2021 年 3 月 14 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇二一年三月十四日”。使用全汉字的日期格式。

`old` 效果为“March 14, 2021”。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。

6.2 文档标题汉化

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

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

`contentsname` ★ `contentsname = <名字>`

设置目录标题名 `\contentsname`。中文默认为“目录”。

`listfigurename` ★ `listfigurename = <名字>`

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

`listtablename` ★ `listtablename = <名字>`

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

`figurename` ★ `figurename = <名字>`

设置图片环境标题名 `\figurename`。中文默认为“图”。

`tablename` ★ `tablename = <名字>`

设置表格环境标题名 `\tablename`。中文默认为“表”。

`abstractname` ★ `abstractname = <名字>`

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

`indexname` ★ `indexname = <名字>`

设置索引标题名 `\indexname`。中文默认为“索引”。

`appendixname` ★ `appendixname = <名字>`

设置附录标题名 `\appendixname`。中文默认为“附录”。

`bibname` ★ `bibname = <名字>`

设置参考文献标题名。中文默认为“参考文献”。

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

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

`proofname` ★ `proofname = <名字>`

设置证明环境的名称 `\proofname`。中文默认为“证明”。

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

`refname` ★ `refname = <名字>`

设置参考文献标题名 `\refname`。中文默认为“参考文献”。

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

`algorithmname` ★ `algorithmname = <名字>`

设置算法环境标题名 `\algorithmname`。中文默认为“算法”。

`continuation` ★ `continuation = <名字>`

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

6.3 页面格式设置与汉化

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

`\pagestyle{headings}`

在页眉中显示当前章节的编号与标题。

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

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

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

例 10

```

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

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

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

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{\CTEXifname{\CTEXthesection——}{\#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第7节 章节标题样式设置

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

本节涉及的所有选项均需使用 \ctexset 命令设置。

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

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

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

多层选项之间用斜线分开, 例如, part/name 选项设置 \part 标题的在数字前后的名称, 而 section/number 选项设置 \section 标题的数字类型。注意, 斜线 / 的前后不能有空格或者换行。

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

例 12

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

7.1 编号相关

part/numbering	★
chapter/numbering	★
section/numbering	★
subsection/numbering	★
subsubsection/numbering	★
paragraph/numbering	★
subparagraph/numbering	★
New: 2015-06-21	

numbering = true|false

控制是否对不带星号的章节标题进行编号。各级标题的默认值均为 true。

L^AT_EX 标准的章节标题命令(如 \section)大体上完成四份工作: 输出标题内容、对标题编号(计数器增加 1)、将标题列入目录(若调用了 hyperref 宏包还会添加 PDF 书签)、更新页眉页脚标记。带星号的章节标题命令(如 \section*)只简单地输出章节标题内容,但不标题编号,不将标题列入目录或 PDF 书签,也不写入页眉页脚标记。与之不同的是,本选项仅仅是否对不带星号的章节标题进行编号。因此,当设置本选项为 false 时,除了不对标题编号以外,其余功能与正常标题一致:可以编入目录,并生成正确的 hyperref 目录超链接位置和页眉页脚标记。例如:

例 13

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

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

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

secnumdepth	★
New: 2020-05-06	

secnumdepth = 〈整数或章节名称〉

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

本选项对 beamer/ctexbeamer 文档类无效。

表 5 章节层次

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

表 6 章节层次的默认设置

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

如果没有特别说明,以下将用“...”代表各级章节标题名。

.../name ★ name = {⟨前名字⟩,⟨后名字⟩}
 name = {⟨前名字⟩}

Updated: 2014-03-08

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

例 14

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

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

表 7 name 选项的默认设置

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

.../number ★ number = {⟨数字输出命令⟩}

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

例 15

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

将会使 `\section` 的编号变为大写罗马数字(如 I、II 等)。

`number` 选项定义的同时将控制对章节计数器的交叉引用。在引用计数器时，记录在 L^AT_EX 辅助文件中的是 `number` 选项的定义。

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

7.2 格式相关

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

表 8 number 选项的默认设置

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

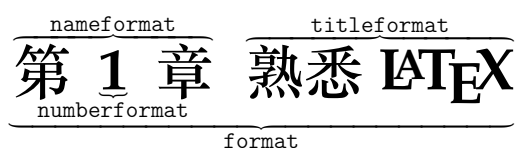


图 1 numberformat, nameformat, titleformat, format 几个选项的作用范围示意

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

Updated: 2020-04-22

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

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

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

例 16

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

format 选项的默认设置见表 9。

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

Updated: 2015-06-30

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

nameformat 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字和编号,以实现特殊效果(见例 22)。

nameformat 选项的默认设置见表 10。

表 9 format 选项的默认设置

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

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

表 10 nameformat 选项的默认设置

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

```
.../numberformat  ★ numberformat = {(格式命令)}
.../numberformat+ ★ numberformat+= {(格式命令)}
```

Updated: 2015-06-19

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

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

例 17

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

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

第 4 章

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

```
.../titleformat  ★ titleformat = {(格式命令)}
.../titleformat+ ★ titleformat+= {(格式命令)}
```

Updated: 2015-06-30

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

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

例 18

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

上面的代码可以做出类似这样的章标题效果:

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

`titleformat` 选项的默认设置见表 11。

```
.../aftername  ★ aftername = {(代码)}
.../aftername+ ★ aftername+= {(代码)}
```

Updated: 2014-03-08

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

表 11 titleformat 选项的默认设置

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

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

New: 2015-06-19

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

表 12 aftername 选项的默认设置

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

表 13 aftertitle 选项的默认设置

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

```
part/pagestyle  * pagestyle = {{页面格式}}
chapter/pagestyle *
```

New: 2014-03-21

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

表 14 pagestyle 选项的默认设置

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

7.3 间距、缩进相关

section/runin ★ runin = true|false

subsection/runin ★

subsubsection/runin ★

paragraph/runin ★

subparagraph/runin ★

New: 2015-06-27

runin 选项只对 `\section` 级以下标题有意义, 用于确定标题与随后的正文是否排在同一段之上。该选项的默认设置见表 15。

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

表 15 runin 选项的默认设置

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

.../hang ★ hang = true|false

Updated: 2020-04-23

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

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

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

表 16 hang 选项的默认设置

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

表 17 indent 选项的默认设置

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

.../indent ★ indent = {⟨缩进间距⟩}

Updated: 2020-04-23

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

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

例 19

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

.../beforeskip ★ beforeskip = {⟨弹性间距⟩}

Updated: 2016-05-10

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

.../afterskip ★ afterskip = {⟨弹性间距⟩}

Updated: 2015-06-27

afterskip 选项控制章节标题与后面下方之间的距离。

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

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

表 18 beforeskip 选项的默认设置

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

表 19 afterskip 选项的默认设置

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

.../fixskip ★

New: 2016-06-03

fixskip = true|false

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

.../break ★

.../break+ ★

New: 2016-09-19

break = {<格式命令>}

break+= {<格式命令>}

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

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

例 20

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

该选项的默认设置见表 20。

表 20 break 选项的默认设置

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

.../afterindent ★

New: 2015-06-27

afterindent = true|false

afterindent 选项用于设置章节标题后首段的缩进。

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

该选项的默认设置见表 21。

表 21 afterindent 选项的默认设置

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

7.4 目录、附录相关

<hr/> <hr/> tocdepth ★	tocdepth = <整数或章节名称>
<hr/> New: 2020-05-06 <hr/>	
	设置对章节标题编入目录的层次数。tocdepth 的值可以是一个整数，也可以是 part, chapter 等名称。层次数与名称的对应关系见表 5。章节层次的默认设置见表 6。
<hr/> <hr/> .../tocline ★	tocline = {(格式定义)}
<hr/> New: 2016-10-25 <hr/>	
	tocline 选项用于定义章节标题在目录文件(.toc)中的格式。(格式定义)有两个参数: 参数 #1 是 part、chapter 等名字, 参数 #2 是标题内容。该选项的默认设置见表 22。

表 22 tocline 选项的默认设置

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

这里 \CTEXnumberline 的意义是,若标题 #1 没有名字,则不输出 \numberline{\CTEXthe#1} 等编号:
 \CTEXifname{\protect\numberline{\csname CTEXthe#1\endcsname}}{}
其中,\CTEXifname 的定义见 7.5 小节。

<hr/> chapter/lofskip ★	lofskip = {(弹性间距)}
<hr/> chapter/lotskip ★	lotskip = {(弹性间距)}
<hr/> New: 2016-10-01 <hr/>	
	lofskip 选项控制插图目录(.lof)中,章之间的插图标题的距离。同样,lotskip 选项控制表格目录(.lot)中,章之间的表格标题的距离。

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

<hr/> appendix/numbering ★	numbering = true false
<hr/> New: 2015-06-21 <hr/>	
	控制是否对附录章(对应 book 与 report)或附录节(对应 article)进行编号,用法与普通章节对应的 numbering 选项相同。该选项默认值为 true。

<hr/> appendix/name ★	name = {(前名字),(后名字)}
<hr/> Updated: 2014-03-08 <hr/>	name = {(前名字)}
	设置附录章(对应 book 与 report)或附录节(对应 article)的名字,用法与普通章节对应的 name 选项相同。

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

该选项的默认设置见表 23。

表 23 appendix/name 选项的默认设置

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

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

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

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

该选项的默认设置见表 24。

表 24 appendix/number 选项的默认设置

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

7.5 辅助命令

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

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。

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

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

New: 2016-09-18

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

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

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

例 21

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


7.6 示例

我们最后举一个稍微复杂的例子,来看看上述选项的综合应用。

例 22

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

本例的设置效果大致如下:

第
1
章

熟悉 L^AT_EX

第 8 节 实用命令

8.1 字号与间距

`\zihao`

`\zihao {<字号>}`

Updated: 2014-03-08

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

`\ziju`

`\ziju {<中文字符宽度的倍数>}`

Updated: 2014-03-28

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。

Updated: 2014-03-27

8.2 中文数字转换

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

表 25 中文字号

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

`\chinese`

Updated: 2016-05-01

`\chinese {〈counter〉}`
`\pagenumbering {chinese}`

`\chinese` 命令与 `\roman` 等命令的用法类似, 作用在一个 `LaTeX` 计数器上, 将计数器的值以中文数字的形式输出。

`\zhnumber`

New: 2014-03-08

`\zhnumber {〈number〉}`

以中文格式输出数字。这里的数字可以是整数、小数和分数。

`\zhdigits`

New: 2014-03-08

`\zhdigits {〈number〉}`

将阿拉伯数字转换为中文数字串。

`\CTEXnumber`

`\CTEXnumber \〈macro〉 {〈number〉}`

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

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

`\CTEXdigits`

`\CTEXdigits \〈macro〉 {〈number〉}`

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

8.3 杂项

`\CTeX` 用于显示 `CTeX` 标志。

第 9 节 *Lua*TeX 下的中文支持方式

在 *Lua*TeX 下, `CTeX` 宏集依赖 `LuaTeX-j`a 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 `LaTeX 2ε` 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 `LuaTeX-j`a 宏包开始支持竖排, 但 `CTeX` 暂不支持竖排。

9.1 *Lua*TeX 下替代字体的设置

`AlternateFont`

Updated: 2020-04-30

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

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

`CharRange`

New: 2014-04-14

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

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

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

例 23

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

```
declarecharrange ★ \ctexset
                    {
                      declarecharrange =
                        {
                          {⟨name1⟩} {⟨character range1⟩} ,
                          {⟨name2⟩} {⟨character range2⟩} ,
                          ...
                        }
                    }
```

New: 2020-04-30

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

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

```
⟨name⟩ = {⟨alternate font name⟩}
⟨name⟩ = {⟨alternate font name⟩} {⟨alternate font features⟩}
```

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

```
clearalternatfont ★ \ctexset
resetalternatfont ★ {
                      clearalternatfont = {⟨family1, family2, ...⟩} ,
                      resetalternatfont = {⟨family1, family2, ...⟩} ,
                      clearalternatfont ,
                      resetalternatfont
                    }
```

New: 2014-04-15

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

第 10 节 C_TE_X 宏集的配置文件

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

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

但在一些情况下,直接修改配置文件仍是必要的,例如:

- 系统没有安装默认设置的字体文件,无法编译。
- 需要经常编译来自其他系统的中文 T_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 宏集的源代码一样,配置文件采用 L^AT_EX3 的语法编写。

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

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

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

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

10.1 修改宏包默认选项

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

例 24

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

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

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

10.2 宏包载入后的配置

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

例 25

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

例 26

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

10.3 配置标题中文翻译

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

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

4.3 节介绍的用于 `fontset` 选项的自定义字库文件,类似于 $\text{CT}_{\text{E}}\text{X}$ 宏集的配置文件,也应该与其他本地配置文件一起保存在本地 TDS 目录树下,并可以配合 `ctexopts.cfg` 等配置文件使用。

第 11 节 对旧版本的兼容性

11.1 $\text{CT}_{\text{E}}\text{X}$ 0.8a 及以前的版本

在 `ctex-kit` 项目成立之前, $\text{CT}_{\text{E}}\text{X}$ 宏包的最后一个版本是 $\text{CT}_{\text{E}}\text{X}$ 0.8a(2007/05/06)。

第 2 版未考虑对这些很早版本的兼容性。

11.2 $\text{CT}_{\text{E}}\text{X}$ 0.9– $\text{CT}_{\text{E}}\text{X}$ 1.0d

在 2009 年在 `ctex-kit` 项目成立后,新增了 $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{TeX}$ 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 $\text{CT}_{\text{E}}\text{X}$ 宏包相对 1.02d 版本(2014/06/09)的兼容性。

第 2 版的 $\text{CT}_{\text{E}}\text{X}$ 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 $\text{CT}_{\text{E}}\text{X}$ 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。

$\text{CT}_{\text{E}}\text{X}$ 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 $\text{CT}_{\text{E}}\text{X}$ 宏包第 2 版中则完全不再支持 `CCT`。

下面这些是在旧版本 $\text{CT}_{\text{E}}\text{X}$ 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。

<code>cs4size</code>	分别相当于 <code>zihao=-4</code> 和 <code>zihao=5</code> ,过时选项。
<code>c5size</code>	

<code>CCT</code>	相关选项已删除。
<code>CCTfont</code>	

<code>indent</code>	<code>indent</code> 和 <code>noindent</code> 什么也不做,过时选项。
<code>noindent</code>	

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

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

`nozhmap` 选项相当于 `zhmap=false`。过时选项。

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

另外,第 2 版 $\text{CT}_{\text{E}}\text{X}$ 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(4.3 节)。

`punct`
`nopunct`

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

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

`cap`
`nocap`

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

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

`space`
`nospace`

新版本宏包 `space` 选项增加真假值参数。(5.3 节)

`nospace` 选项相当于 `space=false`,成为过时选项。

`fancyhdr`

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

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

`hyperref`

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

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

`fntef`

旧版本的 `fntef` 选项用于统一 `CCTfntef` 与 `CJKfntef` 的界面,新版本 \TeX 宏集不再支持 `CCT`,也不再自动载入 `CJKfntef` 或 `xeCJKfntef` 宏包,而仅在其末尾做适当格式调整。

`fntef` 选项过时,因兼容性保留,功能是根据引擎载入 `CJKfntef` (\pdfTeX) 或 `xeCJKfntef` (\XeTeX) 宏包。

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

在调用 `fntef` 宏包选项的同时,旧版本 \TeX 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 \TeX 宏集中已失去意义。此外,在 \pdfTeX 引擎下,用于设置格式的 `\CJKunderdotbasesep` 等宏也被更名为 `\CTEXunderdotbasesep` 等宏。

在新版本中,上述由 `fntef` 衍生的相关命令和环境均被移除。

`\CTEXsetfont`

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

`\CTEXindent`

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

`\CTEXnoindent`

设置 `\parindent=0pt`。过时命令。

`\CTEXsetup`

`\CTEXsetup[⟨选项⟩]{⟨标题⟩}`

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

`\CTEXoptions`

`\CTEXoptions[⟨选项⟩]`

相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

`\Chinese`

`\Chinese{⟨counter⟩}`

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

captiondelimiter

原为 `\CTEXoptions` 命令的选项, 用于控制 `\caption` 编号后面的标点。此选项已过时, 并在新版本的 \TeX 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 27

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

11.3 \TeX 1.02c 以后的 SVN 开发版

\TeX 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。

\TeX 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 \TeX 2.2 之前的版本

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

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

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

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

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

11.5 \TeX 2.4.1 和 2.4.2

part/fixbeforekip
chapter/fixbeforekip

这两个选项已经被删除, 相应功能由新的选项 `fixskip` 提供。

11.6 \TeX 2.5 之前的版本

\TeX 2.5 有一些比较大的变动。

UTF8
GBK

(pdf) \LaTeX 格式下, 文档编码初始值统一设置成 UTF-8。因此, 仍旧使用 GBK 编码的文档, 需要在文档类或宏包选项中显式指定 GBK。

<hr/> \CTEXunderdot \CTEXunderline \CTEXunderdblline \CTEXunderwave \CTEXsout \CTEXxout CTEXfilltwosides <hr/>	不再默认载入 CJKfntef 或 xeCJKfntef 宏包, 同步移除有关命令和环境。若需使用相关宏包, 建议用户使用 \usepackage 命令主动载入。
<hr/> fntef <hr/>	作为兼容性保留, 会视编译引擎载入相应宏包。
<hr/> windows <hr/>	不再支持 Windows XP 系统, 默认要求 Windows 系统有微软雅黑字体。建议 Windows XP 系统的用户及时更新操作系统。若一定要在 Windows XP 中使用, 请使用 2.5 以前的版本。
<hr/> windowsold windowsnew <hr/>	过时字库选项, 作为兼容性保留, 功能是载入 windows 字库。
<hr/> ubuntu <hr/>	改用思源(Noto CJK)和文鼎字库。该字库不再支持 pdfL ^A T _E X 编译。
<hr/> AlternateFont <hr/>	不再支持将替代字体的可选项放在字体名之前的方括号中, 新的语法是将可选项放在字体名之后的花括号之内。

除了以上列出的选项以外, 当用户使用 C_TE_X 系列文档类, 且使用 L^AT_EX 或 upL^AT_EX 编译时, 若用户没有在文档类选项中显式指定 dvips/dvipdfmx/dvisvgm 等驱动选项, 则文档类指定默认驱动为 DVIPDFM_x。

第 12 节 宏集依赖情况与手工安装方法

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

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

- expl3、xparse 和 l3keys2e 宏包。它们属于 l3kernel 和 l3packages 宏集。
- indentfirst 宏包, 属于 tools 宏集。
- everyisel 宏包, 属于 ms 宏集。
- zhnumber 宏包。
- ➡ 以上是各种编译方式都必需的依赖项。
- CJK 宏集。
- CJKpunct 宏包。
- xCJK2uni 宏包。
- zhmetrics 宏包。
- zhmCJK 宏包, 它还依赖
 - iftex 宏包。
 - ltxcmds 宏包。
 - kvoptions 宏包。
 - kvsetkeys 宏包。
 - keyval 宏包, graphics 宏集。
- ➡ 以上是使用 pdfL^AT_EX 或 L^AT_EX + DVIPDFM_x 的编译方式所需要的依赖项, 其中 zhm-CJK 是可选的。

- xeCJK 宏集, 它还依赖
 - xtemplate 宏包, 它属于 l3packages 宏集。
 - fontspec 宏包。
- ➔ 以上是使用 Xe_{La}T_EX 编译时的依赖项。
- luatexja 宏包, 它还依赖
 - adobemapping 宏包。
 - luaotfload 宏包, 它还依赖 lualibs 宏包。
 - luatexbase 宏包, 它还依赖 ctablestack 宏包。
 - atbegshi 宏包。
 - etoolbox 宏包。
 - iftex 宏包。
 - infwarerr 宏包。
 - ltxcmds 宏包。
 - pdftexcmds 宏包。
 - xkeyval 宏包。
- fontspec 宏包。
- ➔ 以上是使用 Lua_{La}T_EX 编译时的依赖项。
- pxeveryset 宏包, 属于 platex-tools 宏集。
- zhmetrics-uptex 宏包。
- ➔ 以上是使用 up_{La}T_EX 编译时的依赖项。

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 C_TE_X 宏集的底层中文支持方式, 那么你需要自行安装该宏包。zhmCJK 的安装较为复杂。我们建议你

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

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

C_TE_X 宏集已被 T_EX Live 和 MiK_TE_X 收录, 若无特别理由, 我们强烈建议用户使用包管理器安装本宏集。

若要手工安装, 请遵循如下步骤:

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

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)
- 曾祥东 (xdzeng96@gmail.com)

- 李泽平 (zepinglee@gmail.com)
- 周宇恺 (muzimuzhi@gmail.com)
- 张瑞熹 (ruixizhang42@gmail.com)

参考文献

- [1] DONALD ERVIN KNUTH. *The T_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 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

2 <*class|ctex>

3 \tl_const:Nx \c__ctex_version_tl

4 { \cs_if_exist_use:cF { ver@ \@currname . \@currentx } { 9999/99/99 } }

5 <*class>

6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl

7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl

8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl

9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl

10 </class>

11 <*ctex>

12 \msg_new:nnnn { ctex } { subpackage-loaded }

13 { Package~`#1'~can~not~be~loaded~with~`ctex'. }

14 {

15 `#1'~is~actually~a~part~of~`ctex'.\\

16 It~is~not~necessary~to~load~it~separately.

17 }

18 \@ifpackageloaded { ctexsize }

19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }

20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }

21 \@ifpackageloaded { ctexheading }

22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }

23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }

24 </ctex>

25 </class|ctex>

26 <*class|style>

27 \RequirePackage { xparse , l3keys2e }

28 </class|style>

29 <*class|ctex>

检查 expl3 和 l3keys2e 的版本。

30 \msg_new:nnnn { ctex } { l3-too-old }

31 { Support~package~`#1'~too~old. }

32 {

33 Please~update~an~up-to-date~version~of~the~bundles\\

34 `l3kernel'~and~`l3packages'\\

35 using~your~TeX~package~manager~or~from~CTAN.

36 }

37 \@ifpackagelater { expl3 } { 2020/07/17 } { }

38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }

39 <*class>

40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }

41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }

42 </class>

\c__ctex_engine_str
\c__ctex_engine_file_str

引擎检查。目前 L^AT_EX3 将 A^pT_EX 识别为 u^pT_EX。

43 \str_const:Nx \c__ctex_engine_str

44 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }

45 \msg_new:nnnn { ctex } { engine-not-supported }

46 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }

47 { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplatex,~or~aplatex. }

48 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }

49 {

50 \str_const:Nx \c__ctex_engine_file_str

51 { ctex-engine- \c__ctex_engine_str .def }

52 }

53 { \msg_critical:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }

54 </class|ctex>

55 <*class|ctex|ctexheading|ctexsize>

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

```

<!ctexsize> 56 \RequirePackage { ctexhook , ctexpatch }
<!ctexheading> 57 \RequirePackage { fix-cm }
58 <!*ctexsize>
<!ctexheading> 59 \RequirePackage { everyysel }

```

14.1 内部函数与变量

临时变量。

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_box
\l__ctex_tmp_dim
60 \tl_clear_new:N \l__ctex_tmp_tl
61 \int_new:N \l__ctex_tmp_int
62 \box_new:N \l__ctex_tmp_box
63 \dim_new:N \l__ctex_tmp_dim
<!ctexheading>

```

在宏包内部使用的键值选项定义、设置命令。

```

\ctex_define_option:n
\ctex_define:n
\ctex_set:n
\ctex_set:nn
64 </!ctexsize>
65 \cs_new_protected:Npn \ctex_define_option:n
66 { \keys_define:nn { ctex / option } }
67 <!*ctexsize>
68 \cs_new_protected:Npn \ctex_define:n
69 { \keys_define:nn { ctex } }
70 \cs_new_protected:Npn \ctex_set:n
71 { \keys_set:nn { ctex } }
72 \cs_new_protected:Npn \ctex_set:nn #1
73 { \keys_set:nn { ctex / #1 } }
74 </!ctexsize>

```

输入文件, 关闭 L^AT_EX3 语法环境, 并设置 @ 为字母类、^ 为上标和 \endlinechar 为 13。

```

\ctex_file_input:n
75 \cs_new_protected:Npn \ctex_file_input:n #1
76 {
77   \ctex_push_file:
78   \file_input:n {#1}
79   \ctex_pop_file:
80 }
81 \cs_new_protected:Npn \ctex_push_file:
82 {
83   \seq_gpush:Nx \g__ctex_file_status_seq
84   {
85     {
86       \bool_if:NTF \l__kernel_expl_bool
87       { \ExplSyntaxOn }
88       { \ExplSyntaxOff }
89     }
90     { \char_value_catcode:n { 64 } }
91     { \char_value_catcode:n { 94 } }
92     { \int_use:N \tex_endlinechar:D }
93   }
94   \ExplSyntaxOff
95   \char_set_catcode_letter:n { 64 }
96   \char_set_catcode_math_superscript:n { 94 }
97   \int_set:Nn \tex_endlinechar:D { 13 }
98 }
99 \cs_new_protected:Npn \ctex_pop_file:
100 {
101   \seq_gpop:NN \g__ctex_file_status_seq \l__ctex_file_status_tl
102   \exp_after:wN \__ctex_pop_file_aux:nnnn \l__ctex_file_status_tl
103 }
104 \cs_new_protected:Npn \__ctex_pop_file_aux:nnnn #1#2#3#4
105 {
106   #1
107   \char_set_catcode:nn { 64 } {#2}

```



```

108 \char_set_catcode:nn { 94 } {#3}
109 \int_set:Nn \tex_endlinechar:D {#4}
110 }
111 \tl_new:N \l__ctex_file_status_tl
112 \seq_new:N \g__ctex_file_status_seq

```

`\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。

```

113 <!*ctexsize>
114 \cs_new_protected:Npn \ctex_scheme_input:n #1
115 {
116   \ctex_push_file:
117   \tl_if_exist:NTF \c__ctex_class_tl
118   {
119     \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }
120     { \file_input:n { ctex-scheme- #1 .def } }
121   }
122   { \file_input:n { ctex-scheme- #1 .def } }
123   \ctex_pop_file:
124 }
125 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

`\g__ctex_section_depth_int` 若大于 3, 则 `\paragraph` 和 `\subparagraph` 标题单独占一行; 若为 3, 则 `\paragraph` 单独占一行。

```

126 </*!beamer>
127 \int_new:N \g__ctex_section_depth_int
128 \int_gset:Nn \g__ctex_section_depth_int { 2 }
129 </*!beamer>

130 </!ctexsize>
131 </class|ctex|ctexheading|ctexsize>
132 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

133 \msg_new:nnnn { ctex } { package-too-old }
134 { Support~package~`#1'~too~old. }
135 {
136   Please~update~an~up-to-date~version~of~the~package~`#1'\
137   using~your~TeX~package~manager~or~from~CTAN.
138 }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

139 \sys_if_output_pdf:TF
140 { \cs_new_eq:NN \ifctexpdf \if_true: }
141 { \cs_new_eq:NN \ifctexpdf \if_false: }

```

`\ctex_if_preamble:TF` 测试是否在 $\text{\LaTeX}2_{\epsilon}$ 的导言区。在宏包内部初始为真, 文档最开始位置再设置为假。注意, 钩子 `\ctex_after_end_preamble:n` 在 `\AtBeginDocument` 之后执行, 可以与 `\@onlypreamble` 的行为一致。

```

142 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
143 \ctex_after_end_preamble:n { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }

```

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

```

144 \cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
145 { \tl_set:Nx #1 { \__ctex_default_ccwd_aux:n {#2} } }
146 \cs_new:Npn \__ctex_default_ccwd_aux:n #1
147 {
148   \exp_not:n {#1}
149   \exp_after:wN \__ctex_default_ccwd_aux:w
150   \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop

```

```

151 }
152 \exp_last_unbraced:NNNNo
153 \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
154 { \tl_if_empty:nT {#2} { \ccwd } }

```

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

```
155 \tl_new:N \g__ctex_encoding_tl
```

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

```
156 \bool_new:N \g__ctex_zhmCJK_bool
```

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

```
157 \tl_new:N \l__ctex_autoindent_tl
```

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

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

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

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

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

```

<class> 160 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 161 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfPackage }

```

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

```

162 <*class>
163 \clist_new:N \g__ctex_std_options_clist
164 </class>

```

对无效选项给出警告。

```

165 \msg_new:nnn { ctex } { invalid-option }
166 { Option~\l_keys_key_tl'~is~invalid~in~current~mode. }
167 \msg_new:nnn { ctex } { invalid-value }
168 { Value~`#1'~is~invalid~for~the~key~\l_keys_key_tl'. }

```

`\ctex_deprecated_option:nn` 对过时选项或命令给出警告。

```

\ctex_set_deprecated_option:n
\ctex_deprecated_command:Nn
169 \cs_new_protected:Npn \ctex_deprecated_option:n
170 { \msg_warning:nnn { ctex } { deprecated-option } }
171 \cs_new_protected:Npn \ctex_set_deprecated_option:n #1
172 {
173   \ctex_deprecated_option:n { Option~`#1'~is~set. }
174   \ctex_set:nn { option } {#1}
175 }
176 \cs_new_protected:Npn \ctex_deprecated_command:Nn #1#2
177 {
178   \msg_warning:nnxx { ctex } { deprecated-command }
179   { \token_to_str:N #1 } { \exp_not:n {#2} }
180 }
181 \msg_new:nnn { ctex } { deprecated-option }
182 { Option~\l_keys_key_tl'~is~deprecated.\\ #1 }
183 \msg_new:nnn { ctex } { deprecated-command }
184 { Command~`#1'~is~deprecated.\\ #2 }
185 </class|ctex>

```

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

```

186 <class|ctex|ctexsize>
187 \int_new:N \g__ctex_font_size_int
188 \int_gset:Nn \g__ctex_font_size_int { -1 }
189 </class|ctex|ctexsize>

```

14.2 宏包选项

```

190 <class|style>
191 \ctex_define_option:n
192 {
193 </class|style>

194 <class|ctex|ctexsize>
195   zihao .choice: ,
196   zihao .value_required:n = true ,
197   zihao / 5 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
198   zihao / -4 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
199   zihao / false .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
<ctexsize> 200 }
201 </class|ctex|ctexsize>
202 <class|ctex>
203   c5size .code:n = { \ctex_set_deprecated_option:n { zihao = 5 } } ,
204   cs4size .code:n = { \ctex_set_deprecated_option:n { zihao = -4 } } ,
205   c5size .value_forbidden:n = true ,
206   cs4size .value_forbidden:n = true ,

```

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

```

207   linespread .fp_set:N = \l__ctex_line_spread_fp ,
208   linespread .initial:n = { \c_nan_fp } ,
209   linespread .value_required:n = true ,

```

autoindent 自动调整段落的首行缩进功能。

```

210   autoindent .choice: ,
211   autoindent .default:n = { true } ,
212   autoindent / true .code:n =
213   {
214     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
215     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
216   } ,
217   autoindent / false .code:n =
218   {
219     \tl_clear:N \l__ctex_autoindent_tl
220     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
221   } ,
222   autoindent / unknown .code:n =
223   {
224     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
225     \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
226   } ,

```

indent 仅为兼容性保留, 已过时。

```

227   indent .code:n =
228   {
229     \ctex_deprecated_option:n
230     {
231       The~functionality~has~been~removed.\\
232       It's~better~to~set~the~heading~styles~via~`afterindent'~option.
233     }

```

```

234     } ,
235     indent .value_forbidden:n = true ,
236     noindent .code:n =
237     {
238         \ctex_deprecated_option:n
239         {
240             The~functionality~has~been~removed.\
241             It's~better~to~set~the~heading~styles~via~`afterindent'~option.
242         }
243     } ,
244     noindent .value_forbidden:n = true ,

```

GBK 文档编码,默认为 UTF-8。
UTF8

```

245     GBK .code:n =
246     {
247         \sys_if_engine_pdftex:TF
248         { \tl_gset:Nn \g__ctex_encoding_tl { GBK } }
249         {
250             \msg_warning:nn { ctex } { invalid-option }
251             \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
252         }
253     } ,
254     UTF8 .code:n = { \tl_gset:Nn \g__ctex_encoding_tl { UTF8 } } ,
255     GBK .value_forbidden:n = true ,
256     UTF8 .value_forbidden:n = true ,

```

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

```

257     fontset .tl_gset:N = \g__ctex_fontset_tl ,
258     nofonts .code:n = { \ctex_set_deprecated_option:n { fontset = none } } ,
259     adobefonts .code:n = { \ctex_set_deprecated_option:n { fontset = adobe } } ,
260     winfonts .code:n = { \ctex_set_deprecated_option:n { fontset = windows } } ,
261     nofonts .value_forbidden:n = true ,
262     winfonts .value_forbidden:n = true ,
263     adobefonts .value_forbidden:n = true ,

264     zhmap .choice: ,
265     zhmap .default:n = { true } ,
266     zhmap / zhmCJK .code:n =
267     {
268         \bool_gset_true:N \g__ctex_zhmCJK_bool
269         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
270     } ,
271     zhmap / true .code:n =
272     {
273         \bool_gset_false:N \g__ctex_zhmCJK_bool
274         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
275     } ,
276     zhmap / false .code:n =
277     {
278         \bool_gset_false:N \g__ctex_zhmCJK_bool
279         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
280     } ,
281     nozhmap .code:n =
282     { \ctex_set_deprecated_option:n { zhmap = false } } ,
283     nozhmap .value_forbidden:n = true ,

```

punct 设置标点符号输出格式。

```

284     punct .tl_set:N = \l__ctex_punct_tl ,
285     punct .default:n = { quanjiao } ,

```

```

286 punct .initial:n = { quanjiao } ,
287 nopunct .code:n = \ctex_set_deprecated_option:n { punct = plain } ,
288 nopunct .value_forbidden:n = true ,

289 space .choices:nn =
290   { true , auto , false }
291   {
292     \exp_args:Nx \ctex_at_end:n
293     { \ctex_set:n { space = \l_keys_choice_tl } }
294   } ,
295 space .default:n = { true } ,
296 nospace .code:n = { \ctex_deprecated_option:nn { space = false } } ,
297 nospace .value_forbidden:n = true ,

298 heading .bool_set:N = \l__ctex_heading_bool ,

299 </class|ctex>
300 <*class|ctex|ctexheading>

301 <!*beamer>
302 sub3section .code:n =
303   { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
304 sub4section .code:n =
305   { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
306 sub3section .value_forbidden:n = true ,
307 sub4section .value_forbidden:n = true ,
308 <!/beamer>

309 scheme .tl_set:N = \l__ctex_scheme_tl ,
310 <*ctexheading>
311 scheme .default:n = { plain } ,
312 scheme .initial:n = { plain }
313 }
314 </ctexheading>
315 <!*ctexheading>
316 scheme .default:n = { chinese } ,
317 scheme .initial:n = { chinese } ,
318 </!ctexheading>

319 </class|ctex|ctexheading>
320 <*class|ctex>

```

cap 和 **nocap** 是过时选项。

```

321 cap .code:n = { \ctex_set_deprecated_option:n { scheme = chinese } } ,
322 nocap .code:n = { \ctex_set_deprecated_option:n { scheme = plain } } ,
323 cap .value_forbidden:n = true ,
324 nocap .value_forbidden:n = true ,

```

以下三项都是过时的兼容选项,它们会载入有关宏包。

```

325 fntef .code:n =
326   {
327     \sys_if_engine_xetex:TF
328     {
329       \ctex_deprecated_option:n { `xeCJKfntef'~package~is~loaded. }
330       \ctex_at_end:n { \RequirePackage { xCJKfntef } }
331     }
332     {
333       \sys_if_engine_pdftex:TF
334       {
335         \ctex_deprecated_option:n { `CJKfntef'~package~is~loaded. }
336         \ctex_at_end:n { \RequirePackage { CJKfntef } }
337       }
338     }
339   }

```

```

338         {
339             \ctex_deprecated_option:n
340             { Furthermore,~option~`fntef'~is~invalid~in~current~mode. }
341         }
342     }
343 },

344 fancyhdr .code:n =
345 {
346     \ctex_deprecated_option:n { `fancyhdr'~package~is~loaded. }
347     \ctex_at_end:n { \RequirePackage { fancyhdr } }
348 },

349 hyperref .code:n =
350 {
351     \ctex_deprecated_option:n { `hyperref'~package~will~be~loaded. }
352     \ctex_at_end:n
353     {
354         \cs_if_exist:NF \hypersetup
355         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
356     }
357     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
358 },
359 }

360 </class>ctex>
361 <*class>ctex|ctexsize>

```

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

```

362 \tl_clear_new:N \l__ctex_tmp_tl
363 \clist_map_inline:nn
364 {
365     10pt , 11pt , 12pt ,
366     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
367 }
368 {
369     \tl_put_right:Nn \l__ctex_tmp_tl
370     {
371         #1 .code:n =
372 <!*class>
373         { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
374 </!class>
375 <*class>
376         {
377             \int_gset:Nn \g__ctex_font_size_int { 2 }
378             \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
379         } ,
380 </class>
381         #1 .value_forbidden:n = true ,
382     }
383 }
384 \exp_args:No \ctex_define_option:n { \l__ctex_tmp_tl }
385 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

386 <*class>
387 \ctex_define_option:n
388 {
389     unknown .code:n =
390     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
391 }
392 </class>

```

载入选项配置文件。

```
<!ctexsize> 393 \ctex_file_input:n { ctexopts.cfg }
394 </class|ctex|ctexsize>
```

处理宏包选项。

```
395 <*class|style>
396 \ProcessKeysOptions { ctex / option }
397 </class|style>
```

pdf_{La}T_EX 下,如果没有显式指定编码为 UTF8,则给出警告信息。

```
398 <*class|ctex>
399 \msg_new:nnn { ctex } { pdftex-utf8 }
400 { UTF8~will~be~used~as~the~default~encoding. }
401 \tl_if_empty:NT \g__ctex_encoding_tl
402 {
403   \sys_if_engine_pdftex:T
404     { \msg_warning:nn { ctex } { pdftex-utf8 } }
405   \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
406 }
407 </class|ctex>
408 <*class>
```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```
409 \int_case:nn { \g__ctex_font_size_int }
410 {
411   { 0 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt } }
412   { 1 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt } }
413 }
```

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

```
414 <*article>
415 \tl_const:Nn \c__ctex_class_tl { article }
416 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
417 \LoadClass { article }
418 </article>
419 <*book>
420 \tl_const:Nn \c__ctex_class_tl { book }
421 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
422 \LoadClass { book }
423 </book>
424 <*report>
425 \tl_const:Nn \c__ctex_class_tl { report }
426 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
427 \LoadClass { report }
428 </report>
429 <*beamer>
430 \tl_const:Nn \c__ctex_class_tl { beamer }
431 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
432 \LoadClass { beamer }
433 </beamer>
434 </class>
```

14.3 特定引擎支持与设置

14.3.1 ctexbackend.cfg

对于 X_{La}T_EX/pdf_{La}T_EX/Lua_{La}T_EX 等默认直接输出 PDF 的编译方式,用户无需为涉及驱动的宏包指定驱动选项。对于 L_AT_EX 和 up_{La}T_EX 等默认不直接输出 PDF 的编译方式,用户则需要指定驱动选项。

由于历史遗留问题,在使用 \LaTeX 或 \upLaTeX 等编译时,大多数涉及驱动的宏包选定的默认输出驱动都是 Dvips 。考虑当前实际使用频率,以及考虑到 $\text{CT}_{\text{E}}\text{X}$ 宏集对中文支持的默认方式,我们在用户使用 $\text{CT}_{\text{E}}\text{X}$ 系列文档类时,将默认的输出驱动改为 $\text{DVIPDFM}_{\text{x}}$ 。

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

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

```

435 <*backend>
436 \begingroup
437 \expandafter\ifx\csname Umathchardef\endcsname\relax
438 \else\expandafter\endgroup\expandafter\endinput\fi
439 \ifodd
440   \expandafter\ifx\csname pdfoutput\endcsname\relax
441   \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
442   \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
443   \def\x#1{%
444     \if\relax\detokenize{#1}\relax
445     \gdef\@classoptionslist{dvi} \fi
446   \else
447     \let\CTEX@add\@ne
448     \@tfor\x:={dvips}{dvi} \do{%
449       \expandafter\in@\expandafter{\expandafter,\x,}{, #1,}%
450       \ifin@ \let\CTEX@add\tw@ \@break@tfor \fi}%
451     \ifodd\CTEX@add \gdef\@classoptionslist{dvi} \fi
452   \fi}
453   \expandafter\x\expandafter{\@classoptionslist}
454 \fi
455 \endgroup
456 </backend>

```

14.3.2 ctex-engine-pdfTeX.def

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

可以使用 LaTeX 2020/10/01 的钩子机制来统一设置。

```

457 <*pdfTeX|upTeX|apTeX>
458 \cs_new_protected:Npn \ctex_set_zhmap:n
459 { \tl_gput_right:Nx \g__ctex_zhmap_tl }
460 \cs_new_protected:Npn \ctex_use_zhmap:
461 { \tl_use:N \g__ctex_zhmap_tl }
462 \cs_if_exist:NTF \ctex_gadd_ltxhook:nn
463 { \ctex_gadd_ltxhook:nn { shipout/firstpage } { \ctex_use_zhmap: } }
464 {
465   \cs_new_protected:Npn \ctex_add_dvi_zhmap:
466   { \AtBeginDvi { \ctex_use_zhmap: } }
467   \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
468   \ctex_at_end_package:nn { atbegshi }
469   {
470     \AtBeginShipoutFirst { \ctex_use_zhmap: }
471     \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
472   }
473 }
474 \tl_new:N \g__ctex_zhmap_tl
475 \@onlypreamble \ctex_set_zhmap:n
476 </pdfTeX|upTeX|apTeX>
477 <*pdfTeX>

```

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

```
478 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
479 { C19 , C10 , C00 , C09 , C40 , C60 }
```

`\ctex_family_cmap:nn` 在 `\DeclareFontFamily` 的 *<loading-settings>* 中给 CJK 字体族加上 CMap。

```
480 \cs_new_protected:Npn \ctex_family_cmap:nn #1#2
481 {
482   \cs_if_free:cF { #1 + #2 }
483   {
484     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
485     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
486   }
487 }
488 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
489 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn
```

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

```
490 \cs_new_protected:Npn \ctex_add_cmap:n #1
491 {
492   \cs_if_free:NF \CJK@plane
493   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
494 }
495 \cs_new_protected:Npn \ctex_add_cmap:Nn #1#2
496 {
497   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
498   #1
499 }
500 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
501 \cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
502 {
503   \tl_set:Nx \l__ctex_tmp_tl { \str_lowercase:n {#2} \CJK@plane }
504   \tex_immediate:D \tex_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
505   \cs_new_protected:Npx #1
506   {
507     \exp_not:N \tex_pdffontattr:D \exp_not:N \tex_font:D
508     { /ToUnicode ~ \int_use:N \tex_pdflastobj:D \c_space_tl 0 ~ R }
509   }
510 }
```

`\DeclareFontFamily` 只在 pdf_{La}T_EX 下加 CMap。如 `cmap` 宏包被引入,则不重复设置。

```
511 \group_begin:
512 \char_set_catcode_other:N \#
513 \sys_if_output_pdf:TF
514 {
515   \group_end:
516   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
517   { \CTEX@Family@CMap {#1} {#2} }
518   {
519     \ctex_at_end_package:nn { cmap }
520     { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
521   }
522   { \ctex_patch_failure:N \DeclareFontFamily }
523 }
524 { \group_end: }
```

首先检查选项,决定是否载入 `zhmCJK` 宏包。

```
525 \if_bool:N \g__ctex_zhmCJK_bool
526 \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhmCJK }
527 \RequirePackage { zhmCJK }
```

不载入 `zhmCJK` 宏包时直接调用 `CJK` 及相关宏包。

```
528 \else:
```

```

529 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
530   { \RequirePackage { CJK } }
531   { \RequirePackage { CJKutf8 } }
532 \RequirePackage { CJKpunct , CJKspace }

```

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

```

533 \cs_new_protected:Npn \ctex_load_zhmap:nnnn #1#2#3#4
534 {
535   \tl_set:Nn \CJKrmdefault {#1}
536   \tl_set:Nn \CJKsfdefault {#2}
537   \tl_set:Nn \CJKttdefault {#3}
538   \ctex_set_zhmap:n { \ctex_file_input:n { ctex-zhmap- #4 .tex } }
539 }
540 \@onlypreamble \ctex_load_zhmap:nnnn

```

zhmCJK 判断结束。

```
541 \fi:
```

\ctex_CJK_input:n
\CJK@input

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。同时要设置 \endlinechar 为 -1。

```

542 \cs_new_protected:Npn \ctex_CJK_input:n #1
543 {
544   \ctex_push_file:
545   \int_set:Nn \tex_endlinechar:D { -1 }
546   \file_input:n {#1}
547   \ctex_pop_file:
548 }
549 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

\ctex_plane_to_utfxvibe:Nn
\CJK@surr

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

```

550 \if_cs_exist:N \CJK@surr
551 \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
552 {
553   \tl_set:Nx \l__ctex_tmp_tl {#2}
554   \int_set:Nn \l__ctex_tmp_int
555     { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
556   \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
557     { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
558     {
559       \int_sub:Nn \l__ctex_tmp_int { 256 }
560       \tl_gset:Nx #1
561       {
562         \int_to_Hex:n
563         { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
564         \int_to_Hex:n
565         { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
566       }
567     }
568 }
569 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
570 \fi:

```

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

```

571 \AtBeginDocument
572 {

```

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

```

573 \str_if_eq:eeF { \l__ctex_punct_tl } { quanjiao }
574 { \punctstyle { \l__ctex_punct_tl } }
575 }

```

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

```

576 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

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。

```

577 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
578 \str_if_eq:onF { \g__ctex_encoding_tl } { UTF8 }
579 {
580   \CJK@makeActive
581   \CJK@loadBinding { standard }
582 }

```

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。注意先使用 `\ctex_update_default_family:` 更新 `\CJKfamilydefault`。

```

583 \exp_args:Nx \ctex_at_end_preamble:n
584 {
585   \exp_not:N \CJK@envStart
586   { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
587   \exp_not:N \CJKtilde
588 }

```

`zhmCJK` 判断结束。

```

589 \fi:

```

`\ctex_auto_ignorespaces:` 保存 `\CJK@@ignorespaces` 的定义, 方便使用。

```

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

```

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

\CJKhook 备用。

```

591 \cs_new_protected:Npn \ctex_ignorespaces_case:N #1
592 {
593   \cs_set_protected:Npn \ctex_set_ignorespaces:
594     { \cs_set_eq:NN \CJK@ignorespaces #1 }
595   \ctex_set_ignorespaces:
596 }
597 \cs_new_protected:Npn \ctex_set_ignorespaces:
598 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }

```

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

```

599 \tl_if_exist:NF \CJKhook { \tl_new:N \CJKhook }
600 \tl_gput_right:Nn \CJKhook { \ctex_set_ignorespaces: }

```

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

```

601 \cs_new_protected:Npn \ctex_punct_set:n #1
602 {
603   \clist_map_inline:Nn \c__ctex_punct_family_clist
604   {
605     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
606     {
607       \cs_set_eq:cc
608       { CJKpunct@ ##1 @spaces }
609       { c__ctex_ #1 ##1 _punct_spaces_tl }
610     }
611   }
612 }
613 \clist_const:Nn \c__ctex_punct_family_clist
614 {
615   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
616   zhsongb , zhheil , zhheib , zhyoub ,
617   zhyahei , zhyaheib , zhpfb , zhpfb
618 }

```

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

```

619 \cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
620 {
621   \cs_if_free:cF { CJKpunct@ #2 @spaces }
622   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
623 }

```

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

```

624 \cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
625 {
626   \clist_map_inline:nn {#1}
627   {
628     \ctex_punct_map_series:nnn { ##1 } { b } {#2}
629     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
630   }
631 }
632 \cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
633 {
634   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
635   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
636   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
637   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
638   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
639   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
640 }

```

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

```

641 \cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
642 {
643   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
644   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
645   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
646   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
647   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
648   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
649 }

```

`\ctex_punct_space:nn` 定义标点的边界信息。
`\ctexspadef`

```

650 \cs_new_protected:Npn \ctex_punct_space:nn #1#2
651 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
652 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

653 \ctex_file_input:n { ctexspa.def }
654 </pdfTeX>

```

14.3.3 ctex-engine-xetex.def

```

655 <*xetex>
656 \RequirePackage { xeCJK }
657 \exp_args:Nx \xeCJKsetup
658 {
659   LoadFandol = false ,
660   PunctStyle = \l__ctex_punct_tl
661 }

```

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`。

```

662 \@ifpackagelater { fontspec } { 2014/05/25 } { }
663 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }
664 </xetex>

```

14.3.4 ctex-engine-luatex.def

```

665 <*luatex>

```

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

```

666 \msg_new:nnn { ctex } { luatexja-loaded }
667 {
668   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
669   Loading~file~`#1'~will~abort!
670 }
671 \@ifpackageloaded { luatexja }
672 { \msg_critical:nnx { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
673 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }
674 \RequirePackage { luatexja }
675 \@ifpackagelater { luatexja } { 2020/04/12 } { }
676 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }
677 \RequirePackage { fontspec }
678 \@ifpackagelater { fontspec } { 2020/02/21 } { }
679 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.3.4.1 LuaTeX-ja 的默认设置

```

680 \ExplSyntaxOff

```

以下设置抄录自 `l1tjdefs.sty`, 略有改动。

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

```

681 \ltjdefcharrange{1}{{"80-"36F, "1E00-"1EFF}
682 \ltjdefcharrange{2}{{"370-"4FF, "1F00-"1FFF}
683 \ltjdefcharrange{3}{%
684   "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
685 \ltjdefcharrange{4}{%
686   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
687   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
688   "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFF, "1B170-"1F0FF,
689   "1F300-"1FFFF, "2000-"206F}
690 \ltjdefcharrange{5}{{"D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
691 \ltjdefcharrange{6}{%
692   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
693   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,
694   "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
695 \ltjdefcharrange{7}{%
696   "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
697   "A960-"A97F, "AC00-"D7FF}
698 \ltjdefcharrange{8}{{"A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}

```

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

```

699 \ltjdefcharrange{9}{%
700   "00B7, "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

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

```

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

```

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

```

703 \@ifpackagelater{luatexja}{2020/08/08}
704 { \ltjsetparameter { autospacing, autoxspacing, differentjfm = paverage } }
705 {
706   \directlua{
707     local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
708     luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
709   }
710   \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
711     xkanjiskip=.25\zw plus 1pt minus 1pt,
712     autospacing, autoxspacing, jacharrange={-1},
713     yalbaselineshift=\z@, yjabaselineshift=\z@,
714     jcharwidowpenalty=500, differentjfm=paverage
715   }
716 }
717 \ExplSyntaxOn

```

14.3.4.2 LuaTeX-j_a 的补丁

```
718 <@@=ctex_ltj>
```

分组中的字符都是 ALchar 类。

```

\CTEX@alchar
\CTEX@beginallalchar
\CTEX@endallalchar
719 \cs_new_protected:Npn \CTEX@alchar #1
720 { \CTEX@beginallalchar #1 \CTEX@endallalchar }
721 \cs_new_protected:Npn \CTEX@beginallalchar
722 {
723   \group_begin:
724     \ctex_ltj_zero_globaldefs:
725     \ltj@allalchar
726 }
727 \cs_new_protected:Npn \CTEX@endallalchar
728 { \group_end: }

```


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

```

729 \cs_new_protected:Npn \CTEX@chardef@text@cmd #1
730 {
731   \cs_set_eq:NN \@ifdefinable \@@ifdefinable
732   \tl_set:Nn \l__ctex_ltj_cmd_tl {#1}
733   \tex_afterassignment:D \__ctex_ltj_chardef_text_cmd:
734   \tex_chardef:D #1
735 }
736 \tl_new:N \l__ctex_ltj_cmd_tl
737 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd:
738 { \exp_after:wN \__ctex_ltj_chardef_text_cmd_aux:N \l__ctex_ltj_cmd_tl }
739 \cs_new_protected:Npn \__ctex_ltj_chardef_text_cmd_aux:N #1
740 {
741   \int_compare:nNnF {#1} < { "80 }
742   { \cs_set_protected:Npx #1 { \CTEX@alchar { \tex_Uchar:D #1 } } }
743 }

```

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

```

744 \cs_new_protected:Npn \CTEX@text@composite@x #1#2
745 {
746   \CTEX@beginallalchar
747   \cs_if_exist_use:NF #1 {#2}
748   \CTEX@endallalchar
749 }

```

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

```

750 \cs_new_protected:Npx \CTEX@add@unicode@accent #1#2
751 {
752   \CTEX@beginallalchar
753   \exp_not:N \tl_if_blank:nTF {#2} { \tex_Uchar:D "A0 ~ } {#2}
754   \exp_not:N \tex_Uchar:D \tex_numexpr:D #1 \scan_stop:
755   \CTEX@endallalchar
756 }

```

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

```

757 \cs_new_protected:Npn \CTEX@patch@text@cmd #1
758 {
759   \exp_args:Nnc \__ctex_ltj_patch_text_cmd:NN #1
760   { \UnicodeEncodingName \token_to_str:N #1 }
761 }
762 \cs_new_protected:Npn \__ctex_ltj_patch_text_cmd:NN #1#2
763 {
764   \cs_set_eq:NN \CTEX@textcmd #2
765   \ctex_preto_cmd:NnnTF \CTEX@textcmd
766   { \ExplSyntaxOff \makeatletter }
767   { \CTEX@beginallalchar }
768   {
769     \ctex_appto_cmd:NnnTF \CTEX@textcmd
770     { \ExplSyntaxOff \makeatletter }
771     { \CTEX@endallalchar }
772     { \cs_set_eq:NN #2 \CTEX@textcmd }
773     { \ctex_patch_failure:N #1 }
774   }
775   { \ctex_patch_failure:N #1 }
776 }

```

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

```

777 \cs_new_protected:Npn \CTEX@patch@tunec
778 {
779   \cs_set_eq:NN \chardef@text@cmd \CTEX@chardef@text@cmd
780   \ctex_file_input:n { tunec.def }
781   \cs_set_eq:NN \@text@composite@x \CTEX@text@composite@x

```

```

782 \cs_set_eq:NN \add@unicode@accent \CTEX@add@unicode@accent
783 \CTEX@patch@text@cmd \textasteriskcentered
784 }
785 \ifpackageloaded { xunicode }
786 { }
787 { \CTEX@patch@tunec }

```

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

```

788 \ctex_at_end_package:nn { xunicode }
789 {
790   \RequirePackage { xunicode-addon }
791   \AtBeginUTFCommand { \CTEX@beginallalchar }
792   \AtEndUTFCommand { \CTEX@endallalchar }
793 }

```

对 `listings` 的补丁是让代码环境支持 `J\Achar` 类。

```

794 \ctex_at_end_package:nn { listings }
795 { \RequirePackage { lltpj-listings } }

```

14.3.4.3 Lua 函数

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

```

796 \group_begin:
797 \char_set_catcode_space:n { 32 }
798 \lua_now:e
799 {
800   local func = lua.get_functions_table()
801   local ltjfont = luatexja.jfont
802   local getattribute = tex.getattribute
803   local set_lua = token.set_lua
804   local scan_int, scan_arg = token.scan_int, token.scan_argument
805   local new_luafunction = luatexbase.new_luafunction

```

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

```

806   local id = new_luafunction("ctex")
807   func[id] = ltjfont.add_kyenc_list
808   set_lua("ctex_ltj_add_kyenc:n", id, "global", "protected")

```

`\ctex_ltj_is_kenc:n` 判断编码是否属于 `jfont`。

```

809   local id = new_luafunction("ctex")
810   func[id] = ltjfont.is_kenc
811   set_lua("ctex_ltj_is_kenc:n", id, "global")

```

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

```

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

```

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

```

812   local id = new_luafunction("ctex")
813   func[id] = function ()
814     local s = scan_arg()
815     local is_braced, is_quoted
816     if s:sub(1,1) == '{' and s:sub(-1)=='}' then

```

```

817         is_braced = true; s = s:sub(2,-2)
818     end
819     if s:sub(1,1) == '"' and s:sub(-1) == '"' then
820         is_quoted = true; s = s:sub(2,-2)
821     end
822     s = s .. ltjfont.print_aftl_address()
823     if is_braced then s = '{...s...}'
824     elseif is_quoted then s = '"...s..."'
825     end
826     tex.sprint(-2, s)
827 end
828 set_lua("ctex_ltj_patch_external_font:n", id, "global")

```

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

```

829     local id = new_luafunction("ctex")
830     func[id] = function ()
831         font.current(getattribute("ltj@curjfont"))
832     end
833     set_lua("ctex_ltj_use_jfont:", id, "global", "protected")

```

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

```

834     local id = new_luafunction("ctex")
835     func[id] = function ()
836         local b = tonumber(scan_arg())
837         local e = tonumber(scan_arg())
838         local alt = scan_arg()
839         local base = scan_arg()
840         ltjfont.set_alt_font_latex(b, e, alt, base)
841     end
842     set_lua("ctex_ltj_set_alt_font:nnnn", id, "global", "protected")

```

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

```

843     local id = new_luafunction("ctex")
844     func[id] = function ()
845         local base = scan_arg()
846         ltjfont.clear_alt_font_latex(base)
847     end
848     set_lua("ctex_ltj_clear_alt_font:n", id, "global", "protected")

```

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

```

849     local id = new_luafunction("ctex")
850     func[id] = function ()
851         local base = scan_arg()
852         local size = scan_arg()
853         ltjfont.output_alt_font_cmd("y", base)
854         ltjfont.pickup_alt_font_a(size)
855     end
856     set_lua("ctex_ltj_pickup_alt_font:nn", id, "global", "protected")

```

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

```

857     local id = new_luafunction("ctex")
858     func[id] = function ()
859         local num = scan_int()
860         local base = scan_arg()
861         ltjfont.pickup_alt_font_b(num, base)
862     end
863     set_lua("__ctex_ltj_pickup_alt_font:Nn", id, "global", "protected")

```

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

```
function luatexja.jfont.does_alt_set(bbase)
  afl_base = alt_font_table_latex[bbase]
  tex.sprint(cat_lp, afl_base and '\\@firstofone' or '\\@gobble')
end
```

luatexja 20200412.0 以后的定义为:

```
function luatexja.jfont.does_alt_set()
  afl_base = alt_font_table_latex[scan_arg()]
  tex.sprint(cat_lp, afl_base and '\\@firstofone' or '\\@gobble')
end

864   local id = new_luafunction("ctex")
865   func[id] = \LuaTeXjversion > 20200412.0
866     and ltjfont.does_alt_set
867   or function ()
868     local base = scan_arg()
869     ltjfont.does_alt_set(base)
870   end
871   set_lua("__ctex_ltj_if_alt_set:nT", id, "global")
```

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

```
872   local id = new_luafunction("ctex")
873   func[id] = function ()
874     tex.set("globaldefs", 0)
875   end
876   set_lua("ctex_ltj_zero_globaldefs:", id, "global", "protected")
877 }
878 \group_end:
```

14.3.4.4 字体切换方式

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

```
879 \cs_new_protected:Npn \ctex_ltj_select_font:
880 {
881   \group_begin: \exp_args:Nnc \group_end:
882   \cs_if_exist_use:NF { \l__ctex_ltj_current_font_tl }
883   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
884 }
885 \tl_new:N \CJK@family
886 \tl_new:N \l__ctex_ltj_current_font_tl
887 \tl_set:Nn \l__ctex_ltj_current_font_tl
888 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }
```

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

```
889 \cs_new_protected:Npn \__ctex_ltj_select_font_aux:
890 {
891   \group_begin:
892   \tl_set_eq:NN \f@encoding \CJK@encoding
893   \tl_set_eq:NN \f@family \CJK@family
894   \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
895   \ctex_ltj_pickup_font:
896   \group_end:
897   \font@name
898   \__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。

```
899   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
```

```

900     { \__ctex_ltj_select_font_aux: }
901   }
902   \cs_new_protected:Npn \__ctex_ltj_push_fontname:n #1
903   {
904     \seq_gpush:No \g__ctex_ltj_fontname_seq { \font@name }
905     \cs_gset_nopar:Npx \font@name {#1}
906   }
907   \cs_new_protected:Npn \__ctex_ltj_pop_fontname:
908   {
909     \seq_gpop:NNT \g__ctex_ltj_fontname_seq \l__ctex_ltj_tmp_tl
910     { \cs_gset_eq:NN \font@name \l__ctex_ltj_tmp_tl }
911   }
912   \seq_new:N \g__ctex_ltj_fontname_seq

```

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

```

913   \cs_new_protected:Npn \ctex_ltj_pickup_font:
914   {
915     \exp_after:wN \cs_if_exist:NF \font@name
916     {
917       \group_begin:
918       \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
919       \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
920       \define@newfont
921       \group_end:
922     }
923   }
924   \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

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

```

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

```

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

```

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

```

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

```

925   \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
926   {
927     \get@external@font
928     \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
929     {
930       \tl_set:Nx \external@font
931       { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
932     }
933     \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。

```

934     \font@name
935     \ctex_ljtj_use_jfont:
936     \use:c { \f@encoding + \f@family }
937     \use:c { \curr@fontshape }
938 }

```

`\ctex_ljtj_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。

```

939 \cs_new_protected_nopar:Npn \ctex_ljtj_subst_font:
940 {
941   \ctex_ljtj_if_alterate_shape_exist:nF { \curr@fontshape }
942   {
943     \group_begin:
944     \tl_set_eq:NN \CJK@family \f@family
945     \cs_if_exist:cF { \l__ctex_ljtj_current_font_tl }
946     {
947       \cs_gset_protected_nopar:Npx \subst@correction
948       {
949         \cs_new_eq:NN
950         \exp_not:c { \l__ctex_ljtj_current_font_tl }
951         \font@name
952       }
953       \group_insert_after:N \group_insert_after:N
954       \group_insert_after:N \subst@correction
955     }
956     \group_end:
957   }
958 }

```

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

```

959 \prg_new_conditional:Npnn \ctex_ljtj_if_alterate_shape_exist:n #1 { T , F , TF }
960 {
961   \__ctex_ljtj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
962   \prg_return_false:
963 }

```

```

964 \cs_new:Npn \__ctex_ljtj_patch_external_font:w #1 ~ at
965 { \ctex_ljtj_patch_external_font:n {#1} ~ at }

```

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

```

966 \cs_new_protected:Npn \ctex_ljtj_select_alterate_font:
967 {
968   \ctex_ljtj_if_alterate_shape_exist:nT { \l__ctex_ljtj_current_shape_tl }
969   {
970     \ctex_ljtj_pickup_alt_font:nn
971     { \l__ctex_ljtj_current_shape_tl } { \f@size }
972   }

```

```

973 }
974 \tl_new:N \l__ctex_ltj_current_shape_tl
975 \tl_set:Nn \l__ctex_ltj_current_shape_tl
976 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

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

```

977 \cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
978 {
979   \cs_if_exist:cF { #1/\f@size }
980   {
981     \group_begin:
982     \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
983     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
984     \ctex_ltj_pickup_font:
985     \group_end:
986     \__ctex_ltj_pop_fontname:
987   }
988 }

```

`\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。

```

989 \cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2
990 {
991   \ltj@@getjfontnumber #1
992   \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
993 }

```

14.3.4.5 数学字体族

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

`\ctex_ltj_if_jfont:nTF` 参数 `#1` 是一个 $\text{\LaTeX 2}_{\epsilon}$ 编码名称或者字体命令。 $\text{\LaTeX 2}_{\epsilon}$ 字体命令的一般形式是:

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

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

```

994 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
995 {
996   \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
997   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
998 }
999 \cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
1000 {#1}

```

`\ctex_ltj_if_jfont_math:NTF` `#1` 是一个形式为 `\M@<encoding>` 的命令,它由 `\DeclareFontEncoding` 的第三个参数来定义。

```

1001 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
1002 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
1003 \group_begin:
1004   \char_set_catcode_other:N M
1005   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
1006     { \ctex_ltj_if_jfont:nTF {#3} }
1007 \group_end:

```

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

```

1008 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
1009 {

```



```

1010 \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
1011 { \ctex_ltj_get_and_define_fonts_ja:nN }
1012 { \ctex_ltj_get_and_define_fonts_al:nN }
1013 {#1} #2
1014 }
1015 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
1016 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
1017 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
1018 {
1019 \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
1020 \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
1021 \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
1022 \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
1023 \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
1024 \ctex_ltj_pickup_font:
1025 \tl_put_right:Nx \math@fonts
1026 {
1027 \ltj@setpar@global
1028 \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
1029 \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
1030 \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
1031 }
1032 }

```

\use@mathgroup
\ctex_ltj_use_math_group:Nn

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

```

1033 \cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
1034 {
1035 \mode_if_math:T
1036 {
1037 \math@bgroup
1038 \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
1039 \ctex_ltj_math_group_hook:
1040 \ctex_ltj_if_jfont_math:NTF #1
1041 { \jfam } { \mathgroup } #2 \scan_stop:
1042 \math@egroup
1043 }
1044 }
1045 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
1046 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

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

```

1047 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
1048 {
1049 \group_begin:
1050 \cs_set_protected:Npn #1 ##1##2##3
1051 { \ltjsetmathletter { ##1 } }
1052 #2
1053 \group_end:
1054 }
1055 \ctex_at_end_package:n { unicode-math }
1056 {
1057 \cs_if_exist:NTF \um_input_math_symbol_table:
1058 {
1059 \ctex_ltj_set_math_letter:NN
1060 \um_sym:nnn
1061 \um_input_math_symbol_table:
1062 }
1063 {
1064 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
1065 \cs_set_protected:Npn \ctex_ltj_math_group_hook:
1066 { \__um_switchto_literal: }
1067 \ctex_ltj_set_math_letter:NN

```

```

1068         \__um_sym:nnn
1069         \__um_input_math_symbol_table:
1070     }
1071 }

```

14.3.4.6 字体族的定义与使用

LuaTeX-j_a 中与标点格式 plain 对应的 JFM 是 mono。

```

\ctex_mono_jfm:n
\l__ctex_ltj_jfm_tl
1072 \cs_new_protected:Npn \ctex_mono_jfm:n #1
1073 {
1074     \str_if_eq:nnTF {#1} { plain }
1075     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
1076     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
1077 }
1078 \tl_new:N \l__ctex_ltj_jfm_tl
1079 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
1080 \ctex_mono_jfm:o { \l__ctex_punct_tl }

```

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

```

1081 \tl_const:Nn \CJK@encoding { LTJY3 }
1082 \DeclareFontEncoding { \CJK@encoding } { } { }
1083 \use:e
1084 {
1085     \exp_not:N \DeclareFontSubstitution
1086     { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1087 }
1088 \ctex_ltj_add_kyenc:n { \CJK@encoding }
1089 \DeclareFontFamily { \CJK@encoding } { song } { }
1090 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1091 { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1092 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1093 { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1094 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
1095 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
1096 { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1097 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
1098 { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1099 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
1100 \jfam \c__ctex_ltj_math_fam_int

```

这是 luatexja-fontspec 中新增的一些字体选项。

```

1101 \newfontfeature { CID } { cid = #1 }
1102 \newfontfeature { JFM } { jfm = #1 }
1103 \newfontfeature { JFM-var } { jfmvar = #1 }

```

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

```

1104 \keys_define:nn { fontspec-preparse-external }
1105 {
1106     NoEmbed .code:n =
1107     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
1108 }
1109 \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。

```

1110 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1111 {
1112     \group_begin:

```

```

1113 \clist_clear:N \l__ctex_ltj_char_range_clist
1114 \prop_clear:N \l__ctex_ltj_alternate_prop
1115 \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1116 \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1117 \clist_set:N \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1118 \ctex_ltj_set_alternate_family:nnF {#1} {#3}
1119 {
1120 \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1121 \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1122 {#1} { \l__ctex_ltj_font_options_clist }
1123 \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1124 \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1125 \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1126 }
1127 \group_end:
1128 }
1129 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1130 \clist_new:N \l__ctex_ltj_font_options_clist

```

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

```

1131 \cs_new_protected:Npn \__ctex_ltj_use_global_options:N #1
1132 {
1133 \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1134 \clist_put_left:Nx #1
1135 { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
1136 }

```

`\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
1137 \prop_new:N \g__ctex_ltj_family_name_prop
1138 \prop_new:N \g__ctex_ltj_family_font_name_prop
1139 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

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

```

1140 \cs_new_protected:Npn \__ctex_ltj_check_family:n #1
1141 {
1142 \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1143 {
1144 \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1145 \cs_undefine:c { \__ctex_ltj_alternate_cs:n {#1} }
1146 \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1147 {
1148 \use:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1149 \cs_undefine:c { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1150 \cs_undefine:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1151 \prop_gremove:Nn \g__ctex_ltj_reset_alternate_prop {#1}
1152 }
1153 \msg_warning:nnxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1154 }
1155 }
1156 \tl_new:N \l__ctex_ltj_tmp_tl
1157 \msg_new:nnn { ctex } { redefine-family }
1158 { Redefining~CJKfamily~\__ctex_ltj_msg_family_map:n {#1}'~(#{2}). }

```

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

```

1159 \cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1160 {
1161 \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
1162 {
1163 \group_begin:
1164 \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1165 \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }

```

```

1166         { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1167         \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1168         { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1169         \__ctex_ltj_set_altername_family:n {#1}
1170         \group_end:
1171     }
1172 }
1173 \tl_new:N \l__ctex_ltj_base_family_tl
1174 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1175 \cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1176 \cs_new_protected:Npn \__ctex_ltj_set_altername_family:n #1
1177 {
1178     \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1179     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1180     \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1181     \cs_if_exist_use:c { \__ctex_ltj_altername_cs:n {#1} }
1182 }
1183 \cs_new:Npn \__ctex_ltj_altername_cs:n #1 { ctex_ltj/altername_family/#1 }

```

\CJKfamily 切换字体。

```

1184 \NewDocumentCommand \CJKfamily { m }
1185 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1186 \cs_new_protected:Npn \ctex_ltj_switch_family:n #1
1187 {
1188     \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1189     {
1190         \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1191         \selectfont
1192     }
1193     { \__ctex_ltj_family_unknown_warning:n {#1} }
1194 }
1195 \tl_new:N \l__ctex_ltj_family_tl
1196 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

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

```

1197 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:nN #1#2 { T , F , TF }
1198 {
1199     \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1200     { \prg_return_true: }
1201     {
1202         \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1203         {
1204             \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1205             \prg_return_true:
1206         }
1207         { \prg_return_false: }
1208     }
1209 }
1210 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:nN { x } { T , F , TF }

1211 \cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
1212 {
1213     \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1214     {
1215         \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1216         {
1217             \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1218             \msg_warning:nnn { ctex } { family-unknown } {#1}
1219         }
1220     }
1221 }
1222 \seq_new:N \g__ctex_ltj_unknown_family_seq
1223 \msg_new:nnn { ctex } { family-unknown }
1224 {
1225     Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\

```

```

1226   Try~to~use~`\_ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1227 }
1228 \cs_new:Npn \_ctex_ltj_msg_def_family_map:n #1
1229 {
1230   \str_case_e:nnF {#1}
1231   {
1232     \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1233     \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1234     \CJKttdefault { \token_to_str:N \setCJKmonofont }
1235   }
1236   { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1237   [...] \{...\}
1238 }
1239 \cs_new:Npn \_ctex_ltj_msg_family_map:n #1
1240 {
1241   \str_case_e:nnF {#1}
1242   {
1243     \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1244     \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1245     \CJKttdefault { \token_to_str:N \CJKttdefault }
1246   }
1247   {#1}
1248 }

1249 \cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
1250 {
1251   \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1252   { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1253   { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1254   {
1255     \int_gincr:N \g__ctex_ltj_family_int
1256     \_ctex_ltj_fontspec:enn
1257     { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1258     {#1} {#2}
1259   }
1260 }
1261 \cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
1262 { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }
1263 \cs_new_protected:Npn \_ctex_ltj_fontspec:nnn #1#2#3
1264 {
1265   \bool_if:NT \l__ctex_ltj_add_alternate_bool
1266   {
1267     \cs_if_free:cF
1268     { \_ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1269     {
1270       \cs_gset_eq:cc
1271       { \_ctex_ltj_alternate_cs:n { reset / #1 } }
1272       { \_ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1273       \cs_gset_eq:cc
1274       { \_ctex_ltj_alternate_cs:n { clear / #1 } }
1275       { \_ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1276     }
1277     \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1278   }
1279   \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1280   \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1281   \ctex_ltj_switch_family:n {#1}
1282 }
1283 \cs_generate_variant:Nn \_ctex_ltj_fontspec:nnn { e }
1284 \prop_new:N \g__ctex_ltj_fontspec_prop

1285 \cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
1286 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1287 \cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
1288 {
1289   \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1290   {#1} \l__ctex_ltj_tmp_tl

```

```

1291     {
1292       \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1293       {#1} \l__ctex_ltj_font_options_clist
1294       \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1295       \bool_set_true:N \l__ctex_ltj_add_alterate_bool
1296       \ctex_ltj_fontspec:ee
1297       { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1298       { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1299     }
1300     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1301   }
1302   \bool_new:N \l__ctex_ltj_add_alterate_bool
1303   \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1304   \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1305   \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1306   {
1307     \token_to_str:N \addCJKfontfeature (s)~ignored.\
1308     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1309   }

```

为了支持字体属性可选项在前在后两种语法,给出两个辅助工具,自带展开功能。

```

\__ctex_ltj_pass_args:nnnn
1310 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1311 {
1312   \tl_if_novalue:nTF {#2}
1313   { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1314   {
1315     \use:e { #1 {#2} {#3} }
1316     #4
1317   }
1318 }
1319 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1320 {
1321   \use:e { #1 {#4} {#2} }
1322   #3
1323 }

1324 \NewDocumentCommand \setCJKfamilyfont { m o m }
1325 {
1326   \__ctex_ltj_pass_args:nnnn
1327   { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1328   { }
1329 }
1330 \NewDocumentCommand \newCJKfontfamily { o m o m }
1331 {
1332   \tl_set:Nx \l__ctex_ltj_tmp_tl
1333   { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
1334   \cs_new_protected:Npx #2
1335   { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1336   \__ctex_ltj_pass_args:nnnn
1337   { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1338   { }
1339 }
1340 \NewDocumentCommand \CJKfontspec { o m }
1341 {
1342   \__ctex_ltj_pass_args:nnnn
1343   { \ctex_ltj_fontspec:nn {#1} {#2} }
1344   { \tex_ignorespaces:D }
1345 }
1346 \NewDocumentCommand \addCJKfontfeatures { m }
1347 {
1348   \ctex_ltj_add_font_features:x {#1}
1349   \tex_ignorespaces:D
1350 }
1351 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

1352 \NewDocumentCommand \setCJKmainfont { o m }

```

```

1353 {
1354   \__ctex_ltj_pass_args:nnnn
1355   { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
1356   { \normalfont }
1357 }
1358 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1359 \NewDocumentCommand \setCJKsansfont { o m }
1360 {
1361   \__ctex_ltj_pass_args:nnnn
1362   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1363   { \normalfont }
1364 }
1365 \NewDocumentCommand \setCJKmonofont { o m }
1366 {
1367   \__ctex_ltj_pass_args:nnnn
1368   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1369   { \normalfont }
1370 }
1371 \NewDocumentCommand \setCJKmathfont { o m }
1372 {
1373   \__ctex_ltj_pass_args:nnnn
1374   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1375   { }
1376 }
1377 \NewDocumentCommand \defaultCJKfontfeatures { m }
1378 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1379 \clist_new:N \g__ctex_ltj_default_features_clist
1380 \@onlypreamble \setCJKmainfont
1381 \@onlypreamble \setCJKsansfont
1382 \@onlypreamble \setCJKmonofont
1383 \@onlypreamble \setCJKmathfont
1384 \@onlypreamble \setCJKromanfont
1385 \@onlypreamble \defaultCJKfontfeatures

```

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

```

1386 \cs_new_protected:Npn \ctex_ltj_ensure_default_family:
1387 {
1388   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1389   {
1390     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1391     {
1392       \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
1393       { \use:n }
1394       {
1395         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1396         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1397         { \use:n }
1398       }
1399       {
1400         \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1401         {
1402           \prop_map_break:n
1403           { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1404         }
1405       }
1406     }
1407     \normalfont
1408     \ctex_ltj_update_mathfont:
1409   }
1410 }

```

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

```

1411 \cs_new_protected:Npn \ctex_ltj_update_mathfont:
1412 {

```

```

1413 \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1414 { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1415 {
1416 \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1417 { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1418 }
1419 }
1420 \cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
1421 {
1422 \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1423 \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1424 { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1425 \cs_if_free:cTF
1426 { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
1427 {
1428 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1429 { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1430 }
1431 {
1432 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1433 { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
1434 }
1435 }

```

14.3.4.7 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1436 \keys_define:nn { ctex_ltj / fontspec }
1437 {
1438 AlternateFont .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
1439 AlternateFont .value_required:n = true ,
1440 CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1441 CharRange .value_required:n = true
1442 }

```

`\ctex_ltj_set_alternate_prop:n`

保存替代字体序列。

```

1443 \cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
1444 { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
1445 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
1446 {
1447 \clist_set:Nx \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
1448 \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
1449 \exp_args:No \__ctex_ltj_push_alternate_prop:nn
1450 { \l__ctex_ltj_tmp_clist } {#1}
1451 }
1452 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
1453 {
1454 \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
1455 \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
1456 }
1457 \clist_new:N \l__ctex_ltj_tmp_clist
1458 \prop_new:N \l__ctex_ltj_alternate_prop

```

`\ctex_ltj_set_alternate_family:nnF`

如果在字体的选项中设置了 CharRange, 则只设置替代字体。

```

1459 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnF
1460 {
1461 \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1462 { \__ctex_ltj_set_family_aux:nnn }
1463 { \__ctex_ltj_set_alternate_family_aux:nnn }
1464 }
1465 \cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
1466 {
1467 \__ctex_ltj_check_family:n {#1}

```



```

1468 \prop_if_empty:NF \l__ctex_ltj_alternate_prop
1469 { \ctex_ltj_save_alternate_seq:cn { \__ctex_ltj_alternate_cs:n {#1} } {#2} }
1470 #3
1471 }
1472 \cs_new_protected:Npn \__ctex_ltj_set_alternate_family_aux:nnn #1#2#3
1473 { \ctex_ltj_set_alternate_family:nn {#1} {#2} }

```

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

```

1474 \cs_new_protected:Npn \ctex_ltj_save_alternate_seq:Nn #1#2
1475 {
1476   \prop_map_inline:Nn \l__ctex_ltj_alternate_prop
1477   { \__ctex_ltj_save_alternate_auxi:w ##2 { } \q_mark #1 {#2} }
1478 }
1479 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxi:w #1#2#
1480 {
1481   \tl_if_blank:nTF {#2}
1482   { \__ctex_ltj_save_alternate_auxii:w {#1} }
1483   { \__ctex_ltj_save_alternate_auxii:w {#1} {#2} }
1484 }
1485 \cs_new_protected:Npn \__ctex_ltj_save_alternate_auxii:w #1#2#3 #4 \q_mark #5#6
1486 {
1487   \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
1488   \clist_set:Nn \l__ctex_ltj_alternate_options_clist {#3}
1489   \__ctex_ltj_use_global_options:N \l__ctex_ltj_alternate_options_clist
1490   \tl_if_blank:nTF {#2}
1491   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
1492   {
1493     \tl_set:Nx \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
1494     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}
1495   }
1496   \use:e
1497   {
1498     \ctex_ltj_save_alternate_family:Nnnn \exp_not:N #5
1499     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1500     { \exp_not:o { \l__ctex_ltj_alternate_options_clist } }
1501     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1502   }
1503 }
1504 \clist_new:N \l__ctex_ltj_alternate_options_clist
1505 \cs_generate_variant:Nn \ctex_ltj_save_alternate_seq:Nn { c }

```

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

```

1506 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nn #1#2
1507 {
1508   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1509   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1510   \ctex_ltj_set_alternate_family:coonn
1511   { \__ctex_ltj_alternate_cs:n {#1} }
1512   { \l__ctex_ltj_char_range_clist }
1513   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1514 }
1515 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:Nnnnn #1#2#3#4#5
1516 {
1517   \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1518   { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1519   \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1520 }
1521 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

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

```

1522 \cs_new_protected:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1523 {

```

```

1524 \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1525 \cs_gset_protected:Npx #1
1526 { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1527 }

```

实际定义替代字体族。

```

1528 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1529 {
1530   \group_begin:
1531   \cs_set_eq:NN \CJKfamily \use_none:n
1532   \ctex_ltj_swap_cs:NN
1533   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1534   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1535   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1536   \group_end:
1537 }
1538 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

交换两个控制序列的意义。

```

1539 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1540 {
1541   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1542   \cs_set_eq:NN #1 #2
1543   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1544   \cs_undefine:N \__ctex_ltj_tmp:w
1545 }

```

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

```

1546 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1547 \cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
1548 {
1549   \int_gincr:N \g__ctex_ltj_family_int
1550   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1551 }
1552 \int_new:N \g__ctex_ltj_family_int

```

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

```

1553 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1554 {
1555   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1556   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1557   { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
1558 }

```

与 **LuaTeX-j**a 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。

```

1559 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7
1560 {
1561   \clist_map_inline:Nn #1
1562   {
1563     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1564     {
1565       \ctex_ltj_set_alternate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1566       \l__ctex_ltj_char_range_tl
1567     }
1568     { \ctex_ltj_set_alternate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1569   }

```

```

1570 \__ctex_ltj_save_altername_shape:cn
1571 { \__ctex_ltj_altername_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1572 { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
1573 }

```

\ctex_ltj_set_altername_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。

```

1574 \NewDocumentCommand \ctex_ltj_set_altername_shape:nnn
1575 { m m > { \SplitArgument { 1 } { -> } } m }
1576 { \ctex_ltj_set_altername_shape:nnnn {#1} {#2} #3 }
1577 \cs_new_protected:Npn \ctex_ltj_set_altername_shape:nnnn #1#2#3#4
1578 {
1579   \ctex_ltj_set_altername_shape:e
1580   {
1581     \__ctex_ltj_range_normalization:nn {#3} {#4}
1582     { \CJK@encoding / \exp_not:n {#2} }
1583     { \CJK@encoding / \exp_not:n {#1} }
1584   }
1585 }
1586 \cs_new_protected:Npn \ctex_ltj_set_altername_shape:n #1
1587 {
1588   \ctex_ltj_set_alt_font:nnnn #1
1589   \__ctex_ltj_save_altername_shape:cn
1590   { \__ctex_ltj_altername_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1591   { \ctex_ltj_set_alt_font:nnnn #1 }
1592 }
1593 \cs_generate_variant:Nn \ctex_ltj_set_altername_shape:n { e }

```

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

```

1594 \cs_new_protected:Npn \ctex_ltj_set_altername_shape:nnN #1#2#3
1595 {
1596   \tl_map_inline:Nn #3
1597   {
1598     \ctex_ltj_set_altername_shape:n
1599     {
1600       ##1
1601       { \CJK@encoding/#2 }
1602       { \CJK@encoding/#1 }
1603     }
1604   }
1605 }

```

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

```

1606 \cs_new_protected:Npn \__ctex_ltj_save_altername_shape:Nn #1#2
1607 {
1608   \group_begin:
1609   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1610   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1611   \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }
1612   \group_end:
1613 }
1614 \cs_generate_variant:Nn \__ctex_ltj_save_altername_shape:Nn { c }

```

clearalternamefont
resetalternamefont

清除和重置操作总是全局的。

```

1615 \ctex_define:n
1616 {
1617   clearalternamefont .code:n =
1618   { \clist_map_function:eN {#1} \ctex_ltj_clear_altername_font:n } ,

```

```

1619   resetalternatefont .code:n =
1620   { \clist_map_function:eN {#1} \ctex_ltj_reset_alternate_font:n } ,
1621   clearalternatefont .default:n = \l_ctex_ltj_family_tl ,
1622   resetalternatefont .default:n = \l_ctex_ltj_family_tl
1623 }
1624 \cs_new_protected:Npn \ctex_ltj_clear_alternate_font:n #1
1625 {
1626   \group_begin:
1627   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1628   {
1629     \cs_if_exist_use:cT
1630     { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1631     {
1632       \prop_gput:Nno \g__ctex_ltj_reset_alternate_prop
1633       {#1} { \l__ctex_ltj_base_family_tl }
1634       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1635       \selectfont
1636     }
1637   }
1638   { \__ctex_ltj_family_unknown_warning:n {#1} }
1639 \group_end:
1640 }
1641 \cs_new_protected:Npn \ctex_ltj_reset_alternate_font:n #1
1642 {
1643   \group_begin:
1644   \prop_gpop:NnNT \g__ctex_ltj_reset_alternate_prop {#1} \CJK@family
1645   {
1646     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1647     \use:c { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1648     \selectfont
1649   }
1650 \group_end:
1651 }
1652 \prop_new:N \g__ctex_ltj_reset_alternate_prop
1653 \cs_generate_variant:Nn \clist_map_function:nN { e }

```

declarecharrange 预先声明字符范围。

```

1654 \ctex_define:n
1655 {
1656   declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
1657   declarecharrange .value_required:n = true
1658 }
1659 \cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
1660 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1661 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
1662 \cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
1663 { \tl_trim_spaces_apply:nN {#1} \ctex_ltj_declare_char_range:nn }

```

#1 是名字, **#2** 是范围。

```

1664 \cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
1665 {
1666   \tl_clear:N \l__ctex_ltj_char_range_tl
1667   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1668   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1669   \ctex_ltj_def_char_range_key:n {#1}
1670   \tl_clear:N \l__ctex_ltj_char_range_tl
1671 }
1672 \tl_new:N \l__ctex_ltj_char_range_tl
1673 \prop_new:N \g__ctex_ltj_char_range_prop

```

预先解释字符区间的意义。

```

1674 \NewDocumentCommand \ctex_ltj_save_char_range:n
1675 { > { \SplitArgument { 1 } { -> } } m }
1676 { \ctex_ltj_save_char_range:nn #1 }

```

\ctex_ltj_declare_char_range:nn
 \g__ctex_ltj_char_range_prop

\ctex_ltj_save_char_range:n

```

1677 \cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
1678 {
1679   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1680   { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
1681 }
1682 \cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
1683 {
1684   \tl_if_novalue:nTF {#2}
1685   {
1686     { \int_eval:n {#1} }
1687     { \int_eval:n {#1} }
1688   }
1689   {
1690     { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
1691     { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
1692   }
1693 }
1694 \int_const:Nn \c__ctex_ltj_range_min_int { "80 }
1695 \int_const:Nn \c__ctex_ltj_range_max_int { \c_max_char_int }

```

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

```

1696 \cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
1697 {
1698   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1699   {
1700     \keys_define:nn { ctex_ltj / fontspec }
1701     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1702   }
1703 }

```

\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。

```

1704 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1705 {
1706   \tl_if_blank:nTF {#2}
1707   { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1708   { \__ctex_ltj_push_alternate_prop:nn {#1} { {#1} #2 } }
1709 }

```

14.3.4.8 其他设置

在抄录环境中禁用 `autospacing` 和 `autoxspacing`。然而, `LuaTeX-jā` 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
}

1710 \AtBeginDocument
1711 {
1712   \ctex_appto_cmd:NnnTF \verbatim@font
1713   { \char_set_catcode_letter:n { 64 } }
1714   { \CTEX@verbatim@font@hook }
1715   { }
1716   { \ctex_patch_failure:N \verbatim@font }

```

```

1717 }
1718 \cs_new_protected:Npn \CTEX@verbatim@font@hook
1719 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

```

`\@@italiccorr` \LaTeX 的倾斜校正也要重新定义。

```

1720 \cs_set_eq:NN \@@italiccorr \/

```

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

```

1721 \cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N
1722 { \ltj@setpar@global \ltjsetkanjiskip }
1723 \cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
1724 { \ltj@setpar@global \ltjsetxkanjiskip }

1725 <@@=ctex>
1726 </luatex>

```

14.3.5 ctex-engine-uptex.def

```

1727 <*uptex|aptex>

```

按 CJK 的命名习惯模拟 `\CJKfamily`。

```

1728 \NewDocumentCommand \CJKfamily { m }
1729 { \kanjifamily {#1} \selectfont }

```

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

```

1730 \DeclareErrorKanjFont {JY2}{zhrm}{m}{n}{10}
1731 \DeclareKanjSubstitution {JY2}{zhrm}{m}{n}
1732 \DeclareKanjSubstitution {JT2}{zhrm}{m}{n}
1733 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1734 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1735 \jfam \symmincho

```

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

```

1736 \ctex_patch_cmd_once:NnnnTF \em
1737 { \ExplSyntaxOff }
1738 { \eminnershape \else \gtfamily \itshape }
1739 { \eminnershape \else \itshape }
1740 { }
1741 {
1742   \ctex_patch_cmd:Nnn \em
1743   { \mcfamily \upshape \else \gtfamily \itshape }
1744   { \eminnershape \else \itshape }
1745 }
1746 \cs_set_nopar:Npn \eminnershape { \upshape }

```

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

```

1747 \cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
1748 {
1749   \DeclareKanjFamily{JY2}{#1}{}
1750   \DeclareKanjFamily{JT2}{#1}{}
1751   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{}
1752   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{}
1753   \tl_if_empty:nF {#3}

```

```

1754     {
1755         \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{ }
1756         \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{ }
1757         \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{ }
1758         \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{ }
1759     }
1760 }

```

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

```

1761 \cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
1762 {
1763     \ctex_set_zhmap:n
1764     {
1765         \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
1766         \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
1767         \tl_if_empty:nF {#3}
1768         {
1769             \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
1770             \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
1771         }
1772     }
1773 }

```

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

```

1774 \cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
1775 {
1776     \ctex_set_zhmap:n
1777     {
1778         \special { pdf:mapline~#1-h~unicode~#2 }
1779         \special { pdf:mapline~#1-v~unicode~#2 }
1780         \tl_if_empty:nF {#3}
1781         {
1782             \special { pdf:mapline~#1b-h~unicode~#3 }
1783             \special { pdf:mapline~#1b-v~unicode~#3 }
1784         }
1785     }
1786 }

```

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

```

1787 \cs_new_protected:Npn \ctex_set_upfonts:nnnnnn #1#2#3#4#5#6
1788 {
1789     \ctex_set_upmap:nnn { upserif } {#1} {#2}
1790     \ctex_set_upmap:nnn { upserifit } {#3} { }
1791     \ctex_set_upmap:nnn { upsans } {#4} {#5}
1792     \ctex_set_upmap:nnn { upmono } {#6} { }
1793 }

```

以下命令只能在导言区使用。

```

1794 \@onlypreamble \ctex_set_upfamily:nnn
1795 \@onlypreamble \ctex_set_upmap:nnn
1796 \@onlypreamble \ctex_set_upmap_unicode:nnn
1797 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

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

```

1798 \RequirePackage { pxeveryysel }
1799 </uptex|aptex>

```

14.3.6 修改主要字体命令

修改 `\rmfamily` 等主要字体命令,使得中文字体能随西文主要字体更新。L^AT_EX 2020-02-02 以后的版本可以直接使用 `\@rmfamilyhook` 等钩子。我们暂时需要处理旧内核的情况,以保持兼容性。

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

```
1800 <*pdf|luatex|uptex|aptex>
<pdf|luatex> 1801 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
```

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

```
1802 \cs_new_protected:Npn \ctex_provide_font_hook:NNN #1#2
1803 {
1804   \exp_args:Nc \__ctex_provide_font_hook_aux:NNNN
1805   { CTEX \cs_to_str:N #2 } #1#2
1806 }
1807 \cs_if_exist:NTF \ctex_gadd_ltxhook:nn
1808 {
1809   \cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
1810   {
1811     \tl_new:N #1
1812     \exp_args:Nx \ctex_gadd_ltxhook:nn { \cs_to_str:N #2 } {#1}
1813   }
1814 }
1815 {
1816   \cs_new_protected:Npn \__ctex_provide_font_hook_aux:NNNN #1#2#3#4
1817   {
1818     \tl_new:N #1
1819     \cs_if_exist:NTF #3
1820     { \tl_gput_right:Nn #3 {#1} }
```

注意此处不能用 `\ctex_patch_cmd:Nnn` 来打补丁,因其会关闭 L^AT_EX3 语法,但 `fontspec` 会在 `\rmfamily` 的命令中相应加入 `\l__fontspec_rmfamily_encoding_tl` 等,导致补丁失败。

```
1821       { \ctex_parse_name:NN \tl_replace_once:Nnn #2 {#4} { #1#4 } }
1822     }
1823   }
1824 \ctex_provide_font_hook:NNN \rmfamily \@rmfamilyhook \selectfont
1825 \ctex_provide_font_hook:NNN \sffamily \@sffamilyhook \selectfont
1826 \ctex_provide_font_hook:NNN \ttfamily \@ttfamilyhook \selectfont
<pdf|luatex> 1827 \ctex_provide_font_hook:NNN \normalfont \@defaultfamilyhook \usefont
```

按 CJK 的命名习惯模拟部分命令,并设置默认字体。

```
1828 \tl_if_exist:NF \CJKfamilydefault
1829 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1830 <*pdf|luatex>
1831 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1832 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1833 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }
1834 \tl_gput_right:Nn \CTEX@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
1835 \tl_gput_right:Nn \CTEX@sffamilyhook { \CJKfamily { \CJKsfdefault } }
1836 \tl_gput_right:Nn \CTEX@ttfamilyhook { \CJKfamily { \CJKttdefault } }
1837 \tl_gput_right:Nn \CTEX@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }
1838 </pdf|luatex>
```

upL^AT_EX 不需要补丁 `\normalfont`,只需要修改 `\kanjifamilydefault`。

```
1839 <*uptex|aptex>
1840 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
1841 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
1842 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhht } }
1843 \tl_gput_right:Nn \CTEX@rmfamilyhook { \kanjifamily { \CJKrmdefault } }
```



```

1844 \tl_gput_right:Nn \CTEX@sffamilyhook { \kanjifamily { \CJKsfdefault } }
1845 \tl_gput_right:Nn \CTEX@ttfamilyhook { \kanjifamily { \CJKttdefault } }
1846 \tl_gset:Nn \kanjifamilydefault { \CJKfamilydefault }
1847 </uptex|aptex>

```

zhmCJK 判断结束。

```
<pdfTeX> 1848 \fi:
```

使修改立刻生效,保证导言区字体族正确。

```
1849 \normalfont
```

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

```
<!pdfTeX> 1850 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

\ctex_update_default_family: 在导言区结束, 如果 \CJKfamilydefault 没有被更改, 则在此时根据西文字体的情况更新 \CJKfamilydefault。xeCJK 已经有这个功能, 不需要再调整。

```

1851 \cs_new_protected:Npn \ctex_update_default_family:
1852 {
1853   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1854   {
1855     \group_begin:
1856     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1857     \tl_gset:Nx \CJKfamilydefault
1858     {
1859       \str_case:onF { \familydefault }
1860       {
1861         { \rmdefault } { \exp_not:N \CJKrmdefault }
1862         { \sfdefault } { \exp_not:N \CJKsfdefault }
1863         { \ttdefault } { \exp_not:N \CJKttdefault }
1864       }
1865       { \CJKfamilydefault }
1866     }
1867     \group_end:
1868   }

```

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 宏包或 up[®]TeX 时, C19rm、JY2rm 等总是有定义的, 不需要确认。

```

<luatex> 1869   \ctex_ljtj_ensure_default_family:
1870   }

```

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

```

1871 \tl_new:N \l__ctex_family_default_init_tl
1872 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1873 \tl_set:Nx \l__ctex_family_default_init_tl
1874 {
1875   \exp_not:N \__ctex_family_default_wrap:n
1876   { \exp_not:o { \CJKfamilydefault } }
1877 }
1878 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1879 </pdfTeX|luatex|uptex|aptex>

```

14.3.7 hyperref 兼容性处理

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

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

```

1880 <*class|ctex>
1881 \@ifpackageloaded { hyperref }
1882 {
1883   \cs_new_protected:Npn \ctex_hypersetup:n #1
1884     { \hypersetup {#1} }
1885 }
1886 {
1887   \cs_new_protected:Npn \ctex_hypersetup:n #1
1888     { \PassOptionsToPackage {#1} { hyperref } }
1889 }
1890 </class|ctex>

```

在 pdfTeX 下使用 GBK 编码, DVIPDFMx 驱动可以直接用它的 `\special` 命令, 其他模式用 `xCJK2uni` 宏包处理。使用 UTF-8 编码时, `CJKutf8` 已经处理了书签问题, 但仍需要设置 `pdfencoding` 为 `unicode`, 目的是在书签的开头写入 BOM (`\376\377`), 提示这是 UTF-16BE 字节流。

```

1891 <*pdf|tex>
1892 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1893 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
1894 {
1895   \ctex_hypersetup:n { CJKbookmarks = true }
1896   \sys_if_output_pdf:TF
1897     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1898     {
1899       \ctex_at_end_package:nn { hyperref }
1900       {
1901         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1902         {
1903           \AtBeginShipoutFirst
1904             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1905           }
1906           { \RequirePackage { xCJK2uni } }
1907         }
1908       }
1909     }
1910   { \ctex_hypersetup:n { pdfencoding = unicode } }
1911 </pdf|tex>

```

X_YTeX 和 LuaTeX 统一设置 `pdfencoding` 为 `unicode`。

```

1912 <*xetex|luatex>
1913 \ctex_hypersetup:n { pdfencoding = unicode }
1914 </xetex|luatex>

```

我们假定 upTeX 使用 DVIPDFMx 驱动输出, 于是使用与 pdfTeX 类似的设置。注意 upTeX 需要使用 UTF8-UTF16 的编码转换。

```

1915 <*uptex|aptex>
1916 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1917 \ctex_at_end_package:nn { hyperref }
1918 { \AtBeginShipoutFirst { \special { pdf:tounicode~UTF8-UTF16 } } }
1919 </uptex|aptex>

1920 <*pdf|tex|xetex|luatex|uptex|aptex>

```

14.3.8 CJKfntef、xeCJKfntef 相关设置

对 pdfTeX 与 XeTeX 引擎, 分别在 CJKfntef、xeCJKfntef 宏包的末尾关闭彩色显式等多余格式。

```

1921 <*pdfTeX>
1922 \ctex_at_end_package:nn { CJKfntef }
1923 {
1924   \normalem
1925   \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1926     { \tl_clear:c { CJK#1color } }
1927 </pdfTeX>
1928 <*xetex>
1929 \ctex_at_end_package:nn { xeCJKfntef }
1930 {
1931   \ifpackagelater { xeCJKfntef } { 2014/11/04 }
1932   {
1933     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1934       { \xeCJKsetup { #1 / format = { } } }
1935   }
1936   {
1937     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1938       { \tl_clear:c { CJK#1color } }
1939   }
1940 </xetex>
1941 <*pdfTeX|xetex>
1942 \clist_map_inline:nn
1943   { underdot , underline , underdblline , underwave , sout , xout }
1944   { \__ctex_clear_fntef_color:n {#1} }
1945 }
1946 </pdfTeX|xetex>

```

14.3.9 \ccwd 的更新

```

1947 \cs_new_protected:Npn \ctex_update_ccwd:
1948 <*pdfTeX|xetex>
1949 {
1950   \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1951   \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1952 }
1953 </pdfTeX|xetex>
1954 <*luatex>
1955 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1956 </luatex>
1957 <*uptex|aptex>
1958 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
1959 </uptex|aptex>
1960 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1961 \cs_new_protected:Npn \ctex_update_ccglue:
1962 <*pdfTeX|xetex>
1963 {
1964   \cs_set_protected:Npn \CJKglue
1965     { \skip_horizontal:N \l__ctex_ccglue_skip }
1966 }
1967 </pdfTeX|xetex>
1968 <*luatex>
1969 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
1970 </luatex>
1971 <*uptex|aptex>
1972 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
1973 </uptex|aptex>
1974 \skip_new:N \l__ctex_ccglue_skip

```

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

```

\ctex_if_ccglue_touched:TF
1975 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1976 {
1977   <*pdfTeX|xetex>
1978   \if_meaning:w \CJKglue \__ctex_ccglue:
1979   \prg_return_false: \else: \prg_return_true: \fi:
1980 </pdfTeX|xetex>
1981 <*luatex>
1982   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1983   { \prg_return_false: } { \prg_return_true: }
1984 </luatex>
1985 <*uptex|aptex>
1986   \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
1987   { \prg_return_false: } { \prg_return_true: }
1988 </uptex|aptex>
1989 }
<pdfTeX|xetex> 1990 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }

\ctex_update_em_unit: 将当前汉字的宽度保存到 \ccwd 中备用。不采用 1em, 因为这时的 1em 实际上来自西文字体的
                      信息, 未必等于汉字的宽度, 这似乎在传统的 .tfm 字体上表现更明显。在 pdfTeX 和 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-jā 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 \zw 作为汉字宽
                      度。upTeX 可以直接使用原生的长度单位 zw。

1991 \cs_new_protected:Npn \ctex_update_em_unit:
<pdfTeX|xetex> 1992 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1993 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1994 { \dim_set:Nn \ccwd { 1zw } }

```

14.3.10 其他

`\ctex_add_to_selectfont:n` `\CTEX@selectfont@hook` `\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 `LuaTeX-jā` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。 `everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。 `pxeveryysel` 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。 `KOMA-Script` 宏包也会进行检查, 我们需要小心处理。

```

1995 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1996 {
1997   \cs_set_protected:Npx \CTEX@selectfont@hook
1998   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1999 }
2000 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
2001 \if_cs_exist:N \@EverySelectfont@Init
2002 \group_begin:
2003   \cs_set:Npn \__ctex_tmp:N #1
2004   {
2005     \tl_set:Nn \l__ctex_tmp_tl {#1}
2006     \cs_new_eq:NN \CTEX@selectfont@save #1
2007     \cs_new_protected:Npn \__ctex_restore_selectfont:
2008     {
2009       \cs_if_free:NF \scr@new@selectfont
2010       {

```

CJK 直接修改 `\selectfont` 和 `pxeveryysel` 的补丁, 会使 `KOMA-Script` 的 `\par@update` 失效。

```

2011 <*pdfTeX|uptex|aptex>
2012 \cs_if_free:NF \par@update

```

```

2013 <*uptex|aptex>
2014 {
2015     \tl_put_right:Nn \@EverySelectfont@Init
2016     { \tl_put_right:Nn #1 { \par@update } }
2017 }
2018 </uptex|aptex>
2019 <*pdfTeX>
2020 { \tl_put_right:Nn #1 { \par@update } }
2021 \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
2022 </pdfTeX>
2023 </pdfTeX|uptex|aptex>
2024 \cs_set_eq:NN \scr@new@selectfont #1
<pdfTeX> 2025 \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
2026 }
2027 \tl_put_left:Nn \@EverySelectfont@Init
2028 { \cs_set_eq:NN #1 \CTEX@selectfont@save }
2029 \cs_undefine:N \__ctex_restore_selectfont:
2030 }
2031 }
2032 \ctex_parse_name:NN \__ctex_tmp:N \selectfont
2033 \exp_last_unbraced:NNo \group_end:
2034 \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
2035 { \ExplSyntaxOff }
2036 { \size@update }
2037 { \CTEX@selectfont@hook \size@update }
2038 { \__ctex_restore_selectfont: }
2039 { \ctex_patch_failure:N \selectfont }
2040 \fi:

```

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

```

2041 <*pdfTeX>
2042 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
2043 </pdfTeX>
2044 <*xetex|luatex|uptex|aptex>
2045 \EverySelectfont { \CTEX@selectfont@hook }
2046 </xetex|luatex|uptex|aptex>

```

Attribute 寄存器 \ltj@curjfont 的初始值是 -1,必须把它设置为一个有效的 font.id,否则编译时会直接退出。

```

2047 <*luatex>
2048 \ctex_add_to_selectfont:n
2049 {
2050     \ctex_ltj_select_font:
2051     \ctex_ltj_select_alternate_font:
2052 }
2053 \tl_set:Nn \CJK@family { song } \selectfont
2054 \tl_clear:N \CJK@family
2055 </luatex>

```

\ctex_update_xkanjiskip: 和 \l__ctex_xkanjiskip_skip 在 upTeX 和 LuaTeX-jan 对 \xkanjiskip 都是即时赋值。单位 zw 与字体相关,因此需要每次 \selectfont 的时候更新一次 \xkanjiskip。如果用户设置过 \xkanjiskip,就不更新。注意,同 TeX 的 \baselineskip 一样,如果在一个段落内多次设置了 \kanjiskip 或 \xkanjiskip,只有最后的设置会影响全段。

```

2056 <*luatex|uptex|aptex>
2057 \cs_new_protected:Npn \ctex_update_xkanjiskip:
2058 {
2059     \skip_if_eq:nnT
<luatex> 2060 { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
<uptex|aptex> 2061 { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
2062 {
2063     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
<luatex> 2064 \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip

```

```

<uptex|aptex> 2065      \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
                2066      }
                2067      }
                2068      \tl_new:N \l__ctex_xkanjiskip_tl
                2069      \tl_set:Nn \l__ctex_xkanjiskip_tl
<luatex> 2070      { .25\zw plus 1pt minus 1pt }
<uptex|aptex> 2071      { .25zw plus 1pt minus 1pt }
                2072      \skip_new:N \l__ctex_xkanjiskip_skip
                2073      \skip_set:Nn \l__ctex_xkanjiskip_skip
<luatex> 2074      { \ltjgetparameter { xkanjiskip } }
<uptex|aptex> 2075      { \tex_xkanjiskip:D }

                2076      \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
                2077      </luatex|uptex|aptex>

```

```

\cht  分别从 .jfm 中读取字符高度、深度和宽度,目前仅考虑横排的情况。
\cdp
\c wd
\ctex_update_kanjisize:
2078 <*luatex>
2079 \dim_new:N \cht
2080 \dim_new:N \cdp
2081 \dim_new:N \c wd
2082 \group_begin:
2083 \char_set_catcode_space:n { 32 }
2084 \lua_now:e
2085 {
2086     local nulltable = { }
2087     local fmt = luatexja.jfont.font_metric_table
2088     local getattribute = tex.getattribute
2089     local setdimen = tex.setdimen
2090     local func = lua.get_functions_table()
2091     local id = luatexbase.new_luafunction("ctex")
2092     func[id] = function ()
2093         local ft = fmt[getattribute("ltj@curjfmt")] or nulltable
2094         local ft = ft and ft.char_type or nulltable
2095         local fk = ft and ft[0] or nulltable
2096         setdimen("cht", fk.height or 0)
2097         setdimen("cdp", fk.depth or 0)
2098         setdimen("c wd", fk.width or ft.zw or 0)
2099     end
2100     token.set_lua("ctex_update_kanjisize:", id, "global", "protected")
2101 }
2102 \group_end:
2103 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2104 </luatex>

```

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

```

2105 \ctex_define:n
2106 {
2107 <*pdf|xtex>
2108     space .choice: ,
2109     space / true .code:n =
<pdf|xtex> 2110     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
<xtex> 2111     { \xeCJKsetup { CJKspace = true } } ,
2112     space / auto .code:n =
<pdf|xtex> 2113     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
<xtex> 2114     { \xeCJKsetup { CJKspace = false } } ,
2115     space / false .code:n =
<pdf|xtex> 2116     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
<xtex> 2117     { \xeCJKsetup { CJKspace = false } } ,
2118     space .default:n = { true } ,
2119     space .initial:n = { auto }
2120 </pdf|xtex>
2121 <*luatex|uptex|aptex>
2122     space .code:n =
2123     { \msg_warning:nn { ctex } { invalid-option } }

```

```

2124 </luatex|uptex|aptex>
2125 }

```

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

```

2126 \ctex_define:n
2127 {
2128     punct .code:n =
2129     {
2130         \tl_set:Nx \l__ctex_punct_tl {#1}
2131         \punctstyle { \l__ctex_punct_tl }
2132         \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2133         \ctex_mono_jfm:o { \l__ctex_punct_tl }
2134         \msg_warning:nn { ctex } { invalid-option }
2135     } ,
2136     punct .default:n = { quanjiao } ,
2137 }
2138 </pdfTeX|xetex|luatex|uptex|aptex>

```

14.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```

<class|ctex> 2139 \ctex_file_input:n { \c__ctex_engine_file_str }

```

14.4 用户设置接口

```

2140 <*class|ctex|ctexheading>
2141 \NewDocumentCommand \ctexset { } { \ctex_set:n }
2142 </class|ctex|ctexheading>

```

\CTEXsetup
\CTEXoptions 过时命令。

```

2143 <*class|ctex>
2144 \NewDocumentCommand \CTEXsetup { +0 { } } > { \TrimSpaces } m }
2145 {
2146     \tl_if_blank:nTF {#1}
2147     { \ctex_deprecated_command:Nn \CTEXsetup { } }
2148     {
2149         \ctex_deprecated_command:Nn \CTEXsetup
2150         { \ctexset {~#2~}{~#1~}~is~set. }
2151         \ctex_set:nn {#2} {#1}
2152     }
2153 }
2154 \NewDocumentCommand \CTEXoptions { +0 { } }
2155 {
2156     \tl_if_blank:nTF {#1}
2157     { \ctex_deprecated_command:Nn \CTEXoptions { } }
2158     {
2159         \ctex_deprecated_command:Nn \CTEXoptions
2160         { \ctexset {~#1~}~is~set. }
2161         \ctex_set:n {#1}
2162     }
2163 }

```

14.5 字距与缩进

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

```

2164 \ctex_define:n

```

```

2165 {
2166   autoindent .choice: ,
2167   autoindent .default:n = { true } ,
2168   autoindent / true .code:n =
2169   {
2170     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2171     \ctex_select_size:
2172   } ,
2173   autoindent / false .code:n =
2174   { \tl_clear:N \l__ctex_autoindent_tl } ,
2175   autoindent / unknown .code:n =
2176   {
2177     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2178     \ctex_select_size:
2179   }
2180 }

```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```

2181 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2182 \cs_new_protected:Npn \ctex_select_size:
2183 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }

```

\ctex_update_size: 在字号变化时更新 `\ccwd`、`\parindent` 和汉字间距。字距为零则恢复正常设置。

```

2184 \cs_new_protected:Npn \ctex_update_size:
2185 {
2186   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2187   {
2188     \ctex_update_stretch:
2189     \ctex_update_parindent:
2190   }
2191   { \ctex_update_ziju: }
2192 }
2193 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2194 \tl_new:N \l__ctex_ziju_tl
2195 \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。

```

2196 \ctex_add_to_selectfont:n
2197 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

linestretch 若行宽不是汉字宽度的整数倍, 自然要求伸展它们之间的差。这里设置的是在此基础上的额外伸展量。初始化为一个汉字的宽度。若设置为 `\maxdimen`, 则禁用此功能。参数的默认单位是汉字的宽度 `\ccwd`。

```

2198 \ctex_define:n
2199 {
2200   linestretch .code:n =
2201   {
2202     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2203     \ctex_select_size:
2204   } ,
2205   linestretch .value_required:n = true
2206 }
2207 \tl_new:N \l__ctex_line_stretch_tl
2208 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

\ctex_update_stretch: 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 ε -**T_EX** 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。

```

2209 \cs_new_protected:Npn \ctex_update_stretch:
2210 {
2211   \ctex_update_em_unit:
2212   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2213   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2214     { \__ctex_update_stretch_auxi: }
2215     { \__ctex_update_stretch_auxii: }
2216 }
2217 \cs_new_protected:Npn \__ctex_update_stretch_auxi:
2218 {
2219   \ctex_if_ccglue_touched:TF
2220     { \ctex_update_ccwd: }
2221     {
2222       \dim_set:Nn \l__ctex_tmp_dim
2223         { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2224       \skip_set:Nn \l__ctex_ccglue_skip
2225         { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2226       \ctex_update_ccglue:
2227     }
2228 }
2229 \cs_new_protected:Npn \__ctex_update_stretch_auxii:
2230 {
2231   \int_set:Nn \l__ctex_tmp_int
2232     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2233   \tex_divide:D \l__ctex_tmp_int \ccwd
2234   \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2235     {
2236       \skip_set:Nn \l__ctex_ccglue_skip
2237         {
2238           \c_zero_dim plus \dim_eval:n
2239             {
2240               ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2241               \l__ctex_tmp_int
2242             }
2243         }
2244     }
2245     { \skip_zero:N \l__ctex_ccglue_skip }
2246   \ctex_update_ccglue:
2247 }

```

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

```

2248 \cs_new_protected:Npn \ctex_update_parindent:
2249 {
2250   \tl_if_empty:NF \l__ctex_autoindent_tl
2251     {
2252       \dim_compare:nNnF \parindent = \c_zero_dim
2253         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2254     }
2255 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2256 \NewDocumentCommand \ziju { m }
2257 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2258 \cs_new_protected:Npn \ctex_ziju:n #1
2259 {
2260   \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2261   \ctex_select_size:
2262 }

```

`\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。

```
2263 \cs_new_protected:Npn \ctex_update_ziju:
2264 {
2265   \ctex_update_em_unit:
2266   \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2267   \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2268   \dim_compare:nNnTF \ccwd > \c_zero_dim
```

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

```
2269 {
2270   \dim_set:Nn \l__ctex_tmp_dim
2271   { \linewidth - \ccwd + \l__ctex_ziju_dim }
2272   \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2273   \tex_divide:D \l__ctex_tmp_int \ccwd
2274   \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。

```
2275   \dim_compare:nNnF \parindent = \c_zero_dim
2276   {
2277     \int_compare:nNnF \l__ctex_tmp_int < 3
2278     { \int_sub:Nn \l__ctex_tmp_int { 2 } }
2279   }
2280   \skip_set:Nn \l__ctex_ccglue_skip
2281   {
2282     \l__ctex_ziju_dim
2283     plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2284     minus \dim_min:n { \dim_abs:n { \l__ctex_ziju_dim } }
2285     { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2286   }
2287 }
2288 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2289 \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 $\text{T}_{\text{E}}\text{X}$ 是伸展还是收缩,之前以“当前行是否还放得下一个汉字”为标准加上或减去伸缩值的做法也未必与实际结果一致,所以只好还是设置为 2\ccwd 。

```
2290   \ctex_update_parindent:
2291 }
2292 \dim_new:N \l__ctex_ziju_dim
```

`\CTEXindent`
`\CTEXnoindent`

过时命令。

```
2293 \NewDocumentCommand \CTEXindent { }
2294 {
2295   \ctex_deprecated_command:Nn \CTEXindent
2296   { \parindent is~set~to~2\ccwd. }
2297   \ctex_update_ccwd:
2298   \dim_set:Nn \parindent { 2 \ccwd }
2299 }
2300 \NewDocumentCommand \CTEXnoindent { }
2301 {
2302   \ctex_deprecated_command:Nn \CTEXnoindent
2303   { \parindent is~set~to~0pt. }
2304   \dim_zero:N \parindent
2305 }
```

14.6 中文数字与日期

```
2306 \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhnumber }
```

```

2307 \RequirePackage { zhnumber }

2308 \cs_new:Npn \chinese { \zhnum_counter:n }
2309 \cs_new_eq:NN \@chinese \@zhnum
2310 \cs_new_eq:NN \Chinese \chinese
2311 \cs_new_eq:NN \CTEXcounter \use_none:n

    给 enumitem 宏包注册 \chinese、\zhnum 和 \zhdig。

2312 \ctex_at_end_package:nn { enumitem }
2313 {
2314   \cs_if_free:NF \AddEnumerateCounter
2315   {
2316     \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2317     \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2318     \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2319   }
2320 }

2321 \NewDocumentCommand \CTEXnumber { m m }
2322 { \protected@edef #1 { \zhnumber {#2} } }
2323 \NewDocumentCommand \CTEXdigits { m m }
2324 { \protected@edef #1 { \zhdigits {#2} } }

2325 \cs_set_eq:NN \CTEXtodayold \today
2326 \ctex_define:n
2327 {
2328   today .choice: ,
2329   today / old .code:n =
2330   { \cs_set_eq:NN \today \CTEXtodayold } ,
2331   today / small .code:n =
2332   {
2333     \cs_set_eq:NN \today \zhtoday
2334     \zhnumsetup { time = Arabic }
2335   } ,
2336   today / big .code:n =
2337   {
2338     \cs_set_eq:NN \today \zhtoday
2339     \zhnumsetup { time = Chinese }
2340   } ,
2341   today / unknown .code:n =
2342   { \msg_error:nxx { ctex } { today-undef } {#1} }
2343 }
2344 \msg_new:nnnn { ctex } { today-undef }
2345 { Today~format~`#1'~is~undefined. }
2346 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

14.7 其他中文标题定义

`\proofname` `\proofname` 未在标准文档类中定义,需要确保它非空。

```

2347 \tl_if_exist:NF \proofname
2348 {
2349   \tl_new:N \proofname
2350   \tl_set:Nn \proofname { Proof }
2351 }

2352 \ctex_define:n
2353 {
2354   contentsname .tl_set:N = \contentsname ,
2355   listfigurename .tl_set:N = \listfigurename ,
2356   listtablename .tl_set:N = \listtablename ,
2357   figurename .tl_set:N = \figurename ,
2358   tablename .tl_set:N = \tablename ,
2359   abstractname .tl_set:N = \abstractname ,
2360   indexname .tl_set:N = \indexname ,
2361   appendixname .tl_set:N = \appendixname ,

```

```

2362     proofname      .tl_set:N = \proofname ,
<article> 2363     bibname        .tl_set:N = \refname
<book|report> 2364     bibname        .tl_set:N = \bibname
2365 <*beamer>
2366     algorithmname .tl_set:N = \algorithmname ,
2367     bibname       .tl_set:N = \bibname ,
2368     refname       .tl_set:N = \refname ,
2369     continuation .tl_set:N = \insertcontinuationtext
2370 </beamer>
2371 }

2372 <*ctex>
2373 \msg_new:nnn { ctex } { ctexbibname }
2374 {
2375     Neither~\token_to_str:N \bibname'~nor~\token_to_str:N \refname'~can~be~found.~\
2376     The~key~\bibname'~will~set~\token_to_str:N \ctexbibname'~to~the~given~value.
2377 }
2378 \tl_if_exist:NTF \insertcontinuationtext
2379 {
2380     \ctex_define:n
2381     {
2382         algorithmname .tl_set:N = \algorithmname ,
2383         bibname       .tl_set:N = \bibname ,
2384         refname       .tl_set:N = \refname ,
2385         continuation .tl_set:N = \insertcontinuationtext
2386     }
2387 }
2388 {
2389     \tl_if_exist:NTF \bibname
2390     { \ctex_define:n { bibname .tl_set:N = \bibname } }
2391     {
2392         \tl_if_exist:NTF \refname
2393         { \ctex_define:n { bibname .tl_set:N = \refname } }
2394         {
2395             \msg_warning:nn { ctex } { ctexbibname }
2396             \ctex_define:n { bibname .tl_set:N = \ctexbibname }
2397         }
2398     }
2399 }
2400 </ctex>

2401 </class|ctex>

```

14.8 中文文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```
2402 <*class|heading>
```

14.8.1 定义标题格式选项

\c__ctex_section_headings_seq 保存 \section 级以下标题名字。

```

2403 <*article|book|report>
2404 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2405 { section , subsection , subsubsection , paragraph , subparagraph }
2406 </article|book|report>

2407 <*article|book|report>
2408 \seq_new:N \c__ctex_headings_seq
2409 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2410 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2411 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2412 </article|book|report>
2413 <*beamer>

```

```

2414 \seq_const_from_clist:Nn \c__ctex_headings_seq
2415 { part , section , subsection }
2416 </beamer>

2417 \cs_new_protected:Npn \__ctex_initial_heading:n #1
2418 {
2419   \tl_new:c { CTEX@pre#1 }
2420   \tl_new:c { CTEX@post#1 }
2421   \tl_const:cx { CTEX@the#1 }
2422   {
2423     \exp_not:c { CTEX@pre#1 }
2424     \exp_not:c { CTEX@the#1 }
2425     \exp_not:c { CTEX@post#1 }
2426   }
2427   \tl_const:cx { CTEX@#1name }
2428   {
2429     \group_begin:
2430     \exp_not:c { CTEX@#1@nameformat }
2431     {
2432       \exp_not:c { CTEX@pre#1 }
2433       \exp_not:N \tl_if_empty:NTF
2434       \exp_not:c { CTEX@#1@numberformat }
2435       { \exp_not:c { CTEX@the#1 } }
2436       {
2437         \group_begin:
2438         \exp_not:c { CTEX@#1@numberformat }
2439         \exp_not:c { CTEX@the#1 }
2440         \group_end:
2441       }
2442       \exp_not:c { CTEX@post#1 }
2443     }
2444     \group_end:
2445   }
2446 }

2447 \cs_new_protected:Npn \__ctex_def_heading_keys:n #1
2448 {
2449   \tl_put_right:Nx \l__ctex_tmp_tl
2450   {
2451     #1 .meta:nn = { ctex / #1 } { #####1 } ,
2452     #1 / name .code:n =
2453     { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2454     #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2455     #1 / beforeskip .tl_set:N = \exp_not:c { CTEX@#1@beforeskip } ,
2456     #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2457     #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2458     #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2459     #1 / numbering .initial:n = true ,
2460     #1 / beforeskip .initial:n = \c_zero_skip ,
2461     #1 / afterskip .initial:n = \c_zero_skip ,
2462     #1 / indent .initial:n = \c_zero_dim ,
2463     #1 / beforeskip .value_required:n = true ,
2464     #1 / afterskip .value_required:n = true ,
2465     #1 / indent .value_required:n = true ,
2466     <*article|book|report>
2467     #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2468     #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2469     #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2470     #1 / hang .initial:n = true ,
2471     #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2472     #1 / tocline .cs_set:Np = \exp_not:c { CTEX@#1@tocline }
2473     \exp_not:n { ##1##2 } ,
2474     \__ctex_plus_key_aux:nn {#1} { break } ,
2475     </article|book|report>
2476     \__ctex_plus_key_aux:nn {#1} { format } ,

```

```

2477     \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2478     \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2479     \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2480     \__ctex_plus_key_aux:nn {#1} { aftername } ,
2481     \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2482   }
2483 }
2484 \cs_new:Npn \__ctex_plus_key_aux:nn #1#2
2485 {
2486   #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2487   #1 / #2 + .code:n =
2488   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2489   #1 / #2 ~ + .code:n =
2490   { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2491 }

```

```

\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 \LaTeX 3 的 `clist` 总是会自动忽略空元素，所以设置 `name={, 章}` 后，第一个元素将会是“章”，必须用空的分组保护空元素：`name={ {}, 章}`，这在使用中有些许不便。我们可以改用 `seq` 或者手写函数解析参数来加以改进。为实现的简单起见，这里用了 `xparse` 的 `\SplitArgument`，它带有参数的长度检查。

```

2492 \NewDocumentCommand \ctex_assign_heading_name:nn
2493 { m > { \SplitArgument { 1 } { , } } +m }
2494 { \__ctex_assign_heading_name:nnn {#1} #2 }
2495 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2496 {
2497   \tl_set:cn { CTEX@pre#1 } {#2}
2498   \tl_if_novalue:nTF {#3}
2499   { \tl_clear:c { CTEX@post#1 } }
2500   { \tl_set:cn { CTEX@post#1 } {#3} }
2501 }

```

```

part/pagestyle
chapter/pagestyle
chapter/lofskip
chapter/lotskip

```

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2502 \group_begin:
2503 <*book|report>
2504 \tl_set:Nn \l__ctex_tmp_tl
2505 {
2506   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2507   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2508   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2509   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2510   chapter / lofskip .initial:n = \c_zero_skip ,
2511   chapter / lotskip .initial:n = \c_zero_skip ,
2512   chapter / lofskip .value_required:n = true ,
2513   chapter / lotskip .value_required:n = true ,
2514 }
2515 </book|report>
2516 <*article|beamer>
2517 \tl_clear:N \l__ctex_tmp_tl
2518 </article|beamer>

```

定义标题键值选项。

```

2519 \seq_map_inline:Nn \c__ctex_headings_seq
2520 {
2521   \__ctex_initial_heading:n {#1}
2522   \__ctex_def_heading_keys:n {#1}
2523 }
2524 \exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }

```

```

\CTEX@heading@format@initial

```

标题格式的一些初始设置，包括恢复默认字体，并禁用自动调整首行缩进，禁止在标题中分页。同时用 `\noindent` 抑制首行缩进并进入水平模式。统一在各级标题的 `format` 选项之前

使用。

```
2525 \cs_new_protected:Npn \CTEX@heading@format@initial
2526 {
2527   \normalfont
2528   \tl_clear:N \l__ctex_autoindent_tl
2529   \int_set:Nn \tex_interlinepenalty:D { 10 000 }
2530   \tex_noindent:D
2531 }
```

`\ctex_indent_box:n` 设置 `\parindent`, 并插入用于产生缩进的盒子, 如果缩进为 0, 就不插入。

```
2532 \cs_new_protected:Npn \ctex_indent_box:n #1
2533 {
2534   \dim_set:Nn \tex_parindent:D {#1}
2535   \__ctex_insert_indent:
2536 }
2537 \cs_new_protected:Npn \__ctex_insert_indent:
2538 {
2539   \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
2540     { \tex_indent:D }
2541 }
2542 \cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n
```

14.8.2 标准标题命令的修改

```
2543 <*article|book|report>
```

`\CTEX@fixtopskip` 修正 `book` 和 `report` 类的 `\part` 和 `\chapter` 标题之前的多余空行。

```
2544 <*book|report>
2545 \cs_new_protected:Npn \CTEX@fixtopskip
2546 {
2547   \CTEX@fixheadingskip
2548   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2549     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2550 }
2551 </book|report>
```

`\CTEX@fixheadingskip` 抑制行间粘连, 修正标题前后的多余间距。事实上, 减掉 `\parskip`, 有一定的风险。如果接下来的内容不会进入水平模式(例如在 `format` 选项中使用 `\hrule` 或者 `\hbox`), $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 就不会加上 `\parskip`。这时候就需要用户把 `\parskip` 加到 `beforeskip` 或者 `afterskip` 作为修正。

```
2552 \cs_new_protected:Npn \CTEX@fixheadingskip
2553 {
2554   \par
2555   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2556   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2557 }
2558 \skip_new:N \l__ctex_heading_skip
2559 \cs_new_protected:Npn \CTEX@setheadingskip
2560 { \skip_set:Nn \l__ctex_heading_skip }
2561 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip
```

`\partmark` 提供 `\partmark`。

```
2562 \ProvideDocumentCommand \partmark { m }
2563 { \markboth { } { } }
```

`\CTEXifname` 用于判断当前标题是否有编号。

```
\CTEX@ifnametrue
\CTEX@ifnamefalse
2564 \cs_new_eq:NN \CTEXifname \use_ii:nn
2565 \cs_new_protected:Npn \CTEX@ifnametrue
2566 { \cs_set_eq:NN \CTEXifname \use_i:nn }
```

```

2567 \cs_new_protected:Npn \CTEX@ifnamefalse
2568 { \cs_set_eq:NN \CTEX@ifname \use_ii:nn }

```

\CTEX@addloflotskip 往插图和表格目录中加入额外间距。如果间距为零,则不加入。

```

2569 <*book|report>
2570 \cs_new_protected:Npn \CTEX@addloflotskip #1
2571 {
2572   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2573   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2574   {
2575     \addtocontents { lof }
2576     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2577   }
2578   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2579   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2580   {
2581     \addtocontents { lot }
2582     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2583   }
2584 }
2585 </book|report>

2586 \cs_new_protected:Npn \CTEX@addtocline #1#2
2587 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

2588 <*article>
2589 \renewcommand\part{%
2590   \if@noskipsec \leavevmode \fi
2591   \par
2592   \CTEX@part@break
2593   % \addvspace{4ex}%
2594   \CTEX@setheadingskip \CTEX@part@beforeskip
2595   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2596   \addvspace \CTEX@headingskip
2597   \ifodd \CTEX@part@afterindent
2598     \@afterindenttrue
2599   \else
2600     \@afterindentfalse
2601   \fi
2602   \secdef\@part\@spart}
2603 </article>
2604 <*book|report>
2605 \renewcommand\part{%
2606   % \if@openright
2607   % \cleardoublepage
2608   % \else
2609   % \clearpage
2610   % \fi
2611   \CTEX@part@break
2612   % \thispagestyle{plain}%
2613   \thispagestyle{\CTEX@part@pagestyle}%
2614   \if@twocolumn
2615     \onecolumn
2616     \@tempswattrue
2617   \else
2618     \@tempswafalse
2619   \fi
2620   % \null\vfil
2621   \CTEX@setheadingskip \CTEX@part@beforeskip
2622   \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2623   \vspace*{\CTEX@headingskip}%
2624   \secdef\@part\@spart}

```



```

2625 </book|report>

2626 <*article>
2627 \def\@part[#1]#2{%
2628   \ifnum \c@secnumdepth >\m@ne
2629     \ifodd \CTEX@part@numbering
2630       \CTEX@ifnametrue
2631       \refstepcounter{part}%
2632   %   \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2633     \else
2634       \CTEX@ifnamefalse
2635       \CTEX@makeanchor{part*}%
2636   %   \addcontentsline{toc}{part}{#1}%
2637     \fi
2638   \else
2639     \CTEX@ifnamefalse
2640     \CTEX@makeanchor{part*}%
2641   %   \addcontentsline{toc}{part}{#1}%
2642     \fi
2643   \CTEX@getttitle{#1}%
2644   \CTEX@addtocline{part}{#1}%
2645   \partmark{#1}%
2646   \begingroup
2647   %   \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2648   \CTEX@heading@format@initial
2649   \CTEX@part@format{%
2650   %   \ifnum \c@secnumdepth >\m@ne
2651   %     \Large\bfseries\partname\nobreakspace\thepart\par\nobreak
2652   %   \fi
2653     \CTEX@headinghang{part}%
2654     {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2655   %   \huge\bfseries #2%
2656     \CTEX@part@titleformat{#2}%
2657   %   \markboth{}{}\par
2658     \CTEX@part@afterttitle}\par
2659   \endgroup
2660   \nobreak
2661   %   \vskip 3ex
2662   \CTEX@setheadingskip \CTEX@part@afterskip
2663   \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2664   \vskip \CTEX@headingskip
2665   \@afterheading}
2666 </article>
2667 <*book|report>
2668 \def\@part[#1]#2{%
2669   \ifnum \c@secnumdepth >-2\relax
2670     \ifodd \CTEX@part@numbering
2671       \CTEX@ifnametrue
2672       \refstepcounter{part}%
2673   %   \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2674     \else
2675       \CTEX@ifnamefalse
2676       \CTEX@makeanchor{part*}%
2677   %   \addcontentsline{toc}{part}{#1}%
2678     \fi
2679     \else
2680       \CTEX@ifnamefalse
2681       \CTEX@makeanchor{part*}%
2682   %   \addcontentsline{toc}{part}{#1}%
2683     \fi
2684     \CTEX@getttitle{#1}%
2685     \CTEX@addtocline{part}{#1}%
2686   %   \markboth{}{}\par
2687     \partmark{#1}%
2688     \begingroup
2689   %   \centering \interlinepenalty \@M \normalfont
2690     \CTEX@heading@format@initial

```

```

2691 \CTEX@part@format{%
2692 % \ifnum \c@secnumdepth >-2\relax
2693 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2694 % \fi
2695 \CTEX@headinghang{part}%
2696 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2697 % \Huge\bfseries #2\par
2698 \CTEX@part@titleformat{#2}%
2699 \CTEX@part@aftertitle}\par
2700 \endgroup
2701 \@endpart}
2702 </book|report>

2703 <*article>
2704 \def\@spart#1{%
2705 \CTEX@ifnamefalse
2706 \CTEX@makeanchor@spart{part*}%
2707 \CTEX@getttitle{#1}%
2708 \begingroup
2709 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2710 \CTEX@heading@format@initial
2711 \CTEX@part@format{%
2712 \CTEX@headinghang{part}}}%
2713 % \huge \bfseries #1\par
2714 \CTEX@part@titleformat{#1}%
2715 \CTEX@part@aftertitle}\par
2716 \endgroup
2717 \nobreak
2718 % \vskip 3ex
2719 \CTEX@setheadingskip \CTEX@part@afterskip
2720 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2721 \vskip \CTEX@headingskip
2722 \@afterheading}
2723 </article>
2724 <*book|report>
2725 \def\@spart#1{%
2726 \CTEX@ifnamefalse
2727 \CTEX@makeanchor@spart{part*}%
2728 \CTEX@getttitle{#1}%
2729 \begingroup
2730 % \centering \interlinepenalty \@M \normalfont
2731 \CTEX@heading@format@initial
2732 \CTEX@part@format{%
2733 \CTEX@headinghang{part}}}%
2734 % \Huge \bfseries #1\par%
2735 \CTEX@part@titleformat{#1}%
2736 \CTEX@part@aftertitle}\par
2737 \endgroup
2738 \@endpart}
2739 </book|report>

2740 <*book|report>
2741 \def\@endpart{%
2742 % \vfil
2743 \CTEX@setheadingskip \CTEX@part@afterskip
2744 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2745 \vskip \CTEX@headingskip
2746 \newpage
2747 \if@twoside
2748 \if@openright
2749 \null
2750 \thispagestyle{empty}%
2751 \newpage
2752 \fi
2753 \fi
2754 \if@tempwa
2755 \twocolumn

```

```

2756 \fi}
2757 </book|report>

```

14.8.2.2 chapter 的标题

```

2758 <*book|report>

2759 \renewcommand\chapter{%
2760 % \if@openright\cleardoublepage\else\clearpage\fi
2761 % \thispagestyle{plain}%
2762 \CTEX@chapter@break
2763 \thispagestyle{\CTEX@chapter@pagestyle}%
2764 \global\@topnum\z@
2765 % \@afterindentfalse
2766 \ifodd \CTEX@chapter@afterindent
2767     \@afterindenttrue
2768 \else
2769     \@afterindentfalse
2770 \fi
2771 \secdef\@chapter\@schapter}

2772 \def\@chapter[#1]#2{%
2773     \ifnum \c@secnumdepth >\m@ne
2774 <*book>
2775     \if@mainmatter
2776 </book>
2777         \ifodd \CTEX@chapter@numbering
2778             \CTEX@ifnametrue
2779             \refstepcounter{chapter}%
2780 %             \typeout{\@chapapp\space\thechapter.}%
2781             \typeout{\CTEXthechapter}%
2782 %             \addcontentsline{toc}{chapter}
2783 %             {\protect\numberline{\thechapter}#1}%
2784         \else
2785             \CTEX@ifnamefalse
2786             \CTEX@makeanchor{\Hy@chapapp*}%
2787 %             \addcontentsline{toc}{chapter}{#1}%
2788         \fi
2789 <*book>
2790     \else
2791         \CTEX@ifnamefalse
2792         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2793 %         \addcontentsline{toc}{chapter}{#1}%
2794     \fi
2795 </book>
2796     \else
2797         \CTEX@ifnamefalse
2798         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2799 %         \addcontentsline{toc}{chapter}{#1}%
2800     \fi
2801     \CTEX@getttitle{#1}%
2802     \CTEX@addtocline{chapter}{#1}%
2803     \chaptermark{#1}%
2804 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2805 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2806     \CTEX@addloflotskip{chapter}%
2807     \if@twocolumn
2808         \@topnewpage[\@makechapterhead{#2}]%
2809     \else
2810         \@makechapterhead{#2}%
2811     \afterheading
2812 \fi}

2813 \def\@schapter#1{%
2814     \CTEX@ifnamefalse
2815     \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2816     \CTEX@getttitle{#1}%

```

```

2817 \if@twocolumn
2818 \topnewpage[\@makeschapterhead{#1}]%
2819 \else
2820 \makeschapterhead{#1}%
2821 \afterheading
2822 \fi}

2823 \def\@makechapterhead#1{%
2824 % \vspace*{50\p@}%
2825 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2826 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2827 \vspace*{\CTEX@headingskip}%
2828 \begingroup
2829 % \parindent \z@ \raggedright \normalfont
2830 \CTEX@heading@format@initial
2831 \CTEX@chapter@format{%
2832 % \ifnum \c@secnumdepth >\m@ne
2833 % \if@mainmatter
2834 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2835 % \fi
2836 % \fi
2837 \CTEX@headinghang{chapter}%
2838 {\CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
2839 % \Huge \bfseries #1\par\nobreak
2840 \CTEX@chapter@titleformat{#1}%
2841 \CTEX@chapter@aftertitle}\par
2842 \endgroup
2843 \nobreak
2844 % \vskip 40\p@
2845 \CTEX@setheadingskip \CTEX@chapter@afterskip
2846 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2847 \vskip \CTEX@headingskip}

2848 \def\@makeschapterhead#1{%
2849 % \vspace*{50\p@}%
2850 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2851 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2852 \vspace*{\CTEX@headingskip}%
2853 \begingroup
2854 % \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
2855 \CTEX@heading@format@initial
2856 \CTEX@chapter@format{%
2857 \CTEX@headinghang{chapter}{}}%
2858 % \Huge \bfseries #1\par\nobreak
2859 \CTEX@chapter@titleformat{#1}%
2860 \CTEX@chapter@aftertitle}\par
2861 \endgroup
2862 \nobreak
2863 % \vskip 40\p@
2864 \CTEX@setheadingskip \CTEX@chapter@afterskip
2865 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2866 \vskip \CTEX@headingskip}

2867 </book|report>

```

14.8.2.3 section 类的标题

\@startsection L^AT_EX 的标准参数是：

$$\{ \langle name \rangle \} \{ \langle level \rangle \} \{ \langle indent \rangle \} \{ \langle before skip \rangle \} \{ \langle after skip \rangle \} \{ \langle style \rangle \} * [\langle al theading \rangle] \{ \langle heading \rangle \}$$

```

2868 \def\@startsection#1#2#3#4#5#6{%
2869 \if@noskipsec \leavevmode \fi
2870 \par
2871 % \@tempskipa #4\relax
2872 % \@afterindenttrue
2873 % \ifdim \@tempskipa <\z@

```

```

2874 % \@tempskipa -\@tempskipa \@afterindentfalse
2875 % \fi
2876 \CTEX@update@sectionformat@n{#1}%
2877 \ifodd \CTEX@afterindent
2878 \@afterindenttrue
2879 \else
2880 \@afterindentfalse
2881 \fi
2882 \if@nobreak
2883 \everypar{}%
2884 \else
2885 % \addpenalty\@secpenalty\addvspace\@tempskipa
2886 \csname CTEX@#1@break\endcsname
2887 \CTEX@setheadingskip{#4}%
2888 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2889 \addvspace \CTEX@headingskip
2890 \fi
2891 \@ifstar
2892 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2893 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}

2894 \def\@secntformat#1{%
2895 % \csname the#1\endcsname\quad}%
2896 \csname CTEX@#1name\endcsname
2897 \csname CTEX@#1@aftername\endcsname}

2898 \def\@sect#1#2#3#4#5#6[#7]#8{%
2899 \ifnum #2>\c@secnumdepth
2900 \CTEX@ifnamefalse
2901 \CTEX@makeanchor@sect{#1*}%
2902 \let\@svsec\@empty
2903 \else
2904 \ifodd \csname CTEX@#1@numbering\endcsname
2905 \CTEX@ifnametrue
2906 \refstepcounter{#1}%
2907 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2908 \else
2909 \CTEX@ifnamefalse
2910 \CTEX@makeanchor{#1*}%
2911 \let\@svsec\@empty
2912 \fi
2913 \fi
2914 \CTEX@getttitle{#7}%
2915 % \@tempskipa #5\relax
2916 % \ifdim \@tempskipa>\z@
2917 \unless \ifodd \CTEX@runin
2918 \begingroup
2919 \CTEX@heading@format@initial
2920 #6{%
2921 % \@hangfrom{\hskip #3\relax\@svsec}%
2922 % \interlinepenalty \@M #8\@par
2923 \CTEX@sectionhang{#3}{\@svsec}%
2924 \csname CTEX@#1@titleformat\endcsname{#8}%
2925 \csname CTEX@#1@aftertitle\endcsname}\par
2926 \endgroup
2927 \csname #1mark\endcsname{#7}%
2928 % \addcontentsline{toc}{#1}{%
2929 % \ifnum #2>\c@secnumdepth \else
2930 % \protect\numberline{\csname the#1\endcsname}%
2931 % \fi
2932 % #7}%
2933 \CTEX@addtocline{#1}{#7}%
2934 \else
2935 \def\@svsechd{%
2936 #6{%
2937 % \hskip #3\relax \@svsec #8
2938 {\CTEX@indentbox{#3}}\@svsec

```

```

2939      \csname CTEX@#1@titleformat\endcsname{#8}%
2940      \csname CTEX@#1@aftertitle\endcsname}%
2941      \csname #1mark\endcsname{#7}%
2942 %      \addcontentsline{toc}{#1}{%
2943 %          \ifnum #2>\c@secnumdepth \else
2944 %              \protect\numberline{\csname the#1\endcsname}%
2945 %              \fi
2946 %              #7}%
2947      \CTEX@addtocline{#1}{#7}}%
2948 \fi
2949 \@xsect{#5}}

2950 \def\@ssect#1#2#3#4#5{%
2951     \CTEX@ifnamefalse
2952     \CTEX@getttitle{#5}%
2953 % \@tempskipa #3\relax
2954 % \ifdim \@tempskipa>\z@
2955     \unless \ifodd \CTEX@runin
2956         \begingroup
2957             \CTEX@heading@format@initial
2958             #4{%
2959 %                 \@hangfrom{\hskip #1}%
2960 %                 \interlinepenalty \@M #5\__ctexpar
2961                 \CTEX@sectionhang{#1}{}%
2962                 \CTEX@titleformat@n{#5}%
2963                 \CTEX@aftertitle}\par
2964             \endgroup
2965     \else
2966 % \def\@svsechd{#4{\hskip #1\relax #5}}%
2967     \def\@svsechd{#4{\CTEX@indentbox{#1}}%
2968         \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2969 \fi
2970 \@xsect{#3}}

2971 \def\@xsect#1{%
2972 % \@tempskipa #1\relax
2973 % \ifdim \@tempskipa>\z@
2974     \unless \ifodd \CTEX@runin
2975         \par \nobreak
2976 % \vskip \@tempskipa
2977         \CTEX@setheadingskip{#1}%
2978         \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2979         \vskip \CTEX@headingskip
2980         \@afterheading
2981     \else
2982         \@nobreakfalse
2983         \global\@noskipsectrue
2984         \everypar{%
2985             \if@noskipsec
2986                 \global\@noskipsecfalse
2987                 {\setbox\z@\lastbox}%
2988                 \clubpenalty\@M
2989                 \begingroup \@svsechd \endgroup
2990                 \unskip
2991 %                 \@tempskipa #1\relax
2992 %                 \hskip -\@tempskipa
2993                 \CTEX@heading@glue{#1}%
2994             \else
2995                 \clubpenalty \@clubpenalty
2996                 \everypar{}%
2997             \fi}%
2998 \fi
2999 \ignorespaces}

```

\CTEX@headinghang 分别用于实现 \part/\chapter 和 \section 类标题的 indent 和 hang 选项。
\CTEX@sectionhang

```
3000 \cs_new_protected:Npn \CTEX@headinghang #1
```

```

3001 {
3002   \ctex_heading_hang:cnn
3003   { CTEX@#1@hang }
3004   { \use:c { CTEX@#1@indent } }
3005 }
3006 \cs_new_protected:Npn \CTEX@sectionhang
3007 { \ctex_heading_hang:Nnn \CTEX@hang }

```

\ctex_heading_hang:Nnn \ctex_hang_from:n hang 选项控制是否采用悬挂缩进,同时设置 \parindent。

```

3008 \cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
3009 {
3010   \dim_set:Nn \tex_parindent:D {#2}
3011   \bool_if:NTF #1
3012   { \ctex_hang_from:n }
3013   { \use:n }
3014   { \__ctex_insert_indent: #3 }
3015 }
3016 \cs_new_protected:Npn \ctex_hang_from:n #1
3017 {
3018   \tex_noindent:D
3019   \hbox_set:Nn \l__ctex_tmp_box {#1}
3020   \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
3021   \box_use_drop:N \l__ctex_tmp_box
3022 }
3023 \cs_generate_variant:Nn \ctex_heading_hang:Nnn { c }

```

\ctex_heading_glue:n \CTEX@heading@glue 如果缩进 #1 长度为零,就不插入水平间距。

```

3024 \cs_new_protected:Npn \ctex_heading_glue:n #1
3025 {
3026   \group_begin:
3027   \skip_set:Nn \l__ctex_heading_skip {#1}
3028   \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
3029   { \skip_horizontal:N \l__ctex_heading_skip }
3030   \group_end:
3031 }
3032 \cs_new_eq:NN \CTEX@heading@glue \ctex_heading_glue:n

```

\CTEX@update@sectionformat@n 在 \@startsection 中设置 \CTEX@titleformat@n 等为相应函数。

```

3033 \cs_new_protected:Npn \CTEX@update@sectionformat@n #1
3034 {
3035   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
3036   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
3037   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
3038   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
3039   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
3040   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
3041 }
3042 \cs_new_eq:NN \CTEX@titleformat@n \use:n
3043 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
3044 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
3045 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
3046 \cs_new_eq:NN \CTEX@hang \c_true_bool
3047 \cs_new_eq:NN \CTEX@runin \c_false_bool

3048 \cs_new:Npn \CTEX@part@tocline #1#2
3049 {
3050   \CTEXifname
3051   { \CTEXthepart \hspace { 1em } }
3052   { }
3053   #2
3054 }
3055 <*book|report>
3056 \cs_new:Npn \CTEX@chapter@tocline #1#2
3057 {

```

```

3058     \CTEXifname
3059     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
3060     { }
3061     #2
3062   }
3063 </book|report>

3064 \cs_new:Npn \CTEXnumberline #1
3065 {
3066   \CTEXifname
3067   { \protect \numberline { \use:c { CTEXthe #1 } } }
3068   { }
3069 }

3070 \int_zero:N \l__ctex_tmp_int
3071 \seq_map_inline:Nn \c__ctex_section_headings_seq
3072 {
3073   \int_incr:N \l__ctex_tmp_int
3074   \cs_gset_protected:cpx {#1}
3075   {
3076     \exp_not:N \@startsection {#1}
3077     { \int_use:N \l__ctex_tmp_int }
3078     { \exp_not:c { CTEX@#1@indent } }
3079     { \exp_not:c { CTEX@#1@beforeskip } }
3080     { \exp_not:c { CTEX@#1@afterskip } }
3081     { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
3082   }
3083   \cs_new:cpn { CTEX@#1@tocline } ##1##2
3084   { \CTEXnumberline { ##1 } ##2 }
3085 }

```

14.8.2.4 附录标题

```

3086 \ctex_define:n
3087 {
3088   appendix .meta:nn = { ctex / appendix } {#1} ,
3089   appendix / name .code:n =
3090   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
3091   appendix / number .tl_set:N = \CTEX@appendix@number ,
3092   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3093   appendix / numbering .initial:n = true
3094 }
3095 \tl_new:N \CTEX@preappendix
3096 \tl_new:N \CTEX@postappendix

3097 \cs_new_eq:NN \CTEX@save@appendix \appendix
3098 \cs_gset_protected:Npn \appendix
3099 {
3100   \CTEX@save@appendix
3101 <*article>
3102   \gdef \CTEX@presection { \CTEX@preappendix }
3103   \gdef \CTEX@thesection { \CTEX@appendix@number }
3104   \gdef \CTEX@postsection { \CTEX@postappendix }
3105   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3106 </article>
3107 <*book|report>
3108   \gdef \CTEX@prechapter { \CTEX@preappendix }
3109   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3110   \gdef \CTEX@postchapter { \CTEX@postappendix }
3111   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3112 </book|report>
3113 }

```


14.8.2.5 设置 **hyperref** 宏包的标题锚点

`\CTEX@makeanchor` 设置超链接跳转锚点, 在 **hyperref** 载入后才有意义。

```
3114 \cs_new_protected:Npn \CTEX@makeanchor #1
3115 { }
```

`\c__ctex_headings_cs_seq` 保存内部标题命令的 **CT_EX** 定义, 用于随后比较。

```
3116 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
3117 { part , spart , sect , ssect }
3118 { part , spart , chapter , schapter , sect , ssect }
3119 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3120 {
3121   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3122   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3123 }
```

`\CTEX@hyperheadinghook` **hyperref** 会重定义内部标题命令, 目的在于为没有编号的标题设置锚点 (这一功能受他的 `implicit` 选项的控制)。我们在上面对标题命令的修改已经包含这一功能, 如果这些标题命令在 **hyperref** 载入之前没有被修改过, 则恢复 **CT_EX** 的定义。

```
3124 \cs_new_protected:Npn \CTEX@hyperheadinghook
3125 {
3126   \group_begin:
3127   \ifHy@implicit
3128     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3129     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3130     {
3131       \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3132       {
3133         \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3134         \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3135       }
3136     }
3137   \else:
3138     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3139     { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3140   \fi:
3141   \group_end:
3142 }

3143 \ctex_at_end_package:nn { hyperref }
3144 {
3145   \cs_gset_protected:Npn \CTEX@makeanchor #1
3146   {
3147     \Hy@MakeCurrentHrefAuto {#1}
3148     \Hy@raisedlink
3149     {
3150       \hyper@anchorstart { \@currentHref }
3151       \hyper@anchorend
3152     }
3153   }
3154   \CTEX@hyperheadinghook
3155 }
```

14.8.2.6 兼容 **nameref** 宏包

`\CTEX@getttitle` 在 **nameref** 载入后才有意义, 与上述 **hyperref** 的处理类似。

```
3156 \cs_new_protected:Npn \CTEX@getttitle #1
3157 { }
3158 \ctex_at_end_package:nn { nameref }
3159 {
3160   \cs_gset_protected:Npn \CTEX@getttitle { \NR@getttitle }
```

```

3161 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3162 {
3163   \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3164   { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3165 }
3166 }

```

14.8.2.7 兼容 titlesec 宏包

我们修改了 `\startsection` 的定义, 它的第四个 (`\before`) 和第五个 (`\after`) 参数的符号不再有特殊意义, 改由相应的选项 `afterindent` 和 `runin` 来控制。

引入 `titlesec` 宏包, 并且未设置它的 `loadonly` 选项时, `titlesec` 会展开 `section` 类标题获取它们的参数, 进行初始设置。我们需要进行一些调整。

`\ctex_titlesec_hook:` `\titleformat` 的设置保存在名为 `\ttl@<section>` 的宏中备用, 它的内容是

```
\ttl@<shape>{<format>}{<label>}{<sep>}{<before>}{<after>}
```

我们这里的 `<shape>` 为 `hang` 或者 `runin`。`\titlespacing` 的设置保存在 `\tts@<section>` 之中, 它的内容是

```
{<left>}{<right>}{<before>}{<after>}{<afterindent>}
```

其中 `<afterindent>` 为 1 或 0, 分别对应是否保留段首缩进。我们需要根据 `CTEX` 的 `runin` 和 `afterindent` 选项调整 `\ttl@<shape>` 和 `<afterindent>`。注意, 由 `\ttl@extract` 得的 `<before>` 和 `<after>` 的值总是非负的, 而 `CTEX` 的 `before` 和 `after` 是可以取负值的, 但我们不打算调整它们了。如果使用了 `titlesec` 的 `indentafter` 等选项, 也不需要调整 `\tts@<section>`。

```

3167 \cs_new_protected:Npn \ctex_titlesec_hook:
3168 {
3169   \@ifpackagewith { titlesec } { explicit }
3170   {
3171     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3172     \__ctex_titlesec_format_explicit:Nn
3173   }
3174   { }
3175   \clist_map_inline:nn
3176   { indentafter , noindentafter , indentfirst , nonindentfirst }
3177   {
3178     \@ifpackagewith { titlesec } { ##1 }
3179     {
3180       \clist_map_break:n
3181       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3182     }
3183     { }
3184   }
3185   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3186 }
3187 \cs_new_protected:Npn \__ctex_titlesec_hook:n #1
3188 {
3189   \__ctex_titlesec_format:n {#1}
3190   \exp_args:Nc \__ctex_titlesec_spacing:Nn { tts@#1 } {#1}
3191 }
3192 \cs_new_protected:Npn \__ctex_titlesec_format:n #1
3193 {
3194   \cs_if_free:cF { ttl@#1 }
3195   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttl@#1 } {#1} }
3196 }
3197 \cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
3198 {

```

```

3199 \tl_set:Nx #1
3200 {
3201   \bool_if:cTF { CTEX@#2@runin }
3202   { \exp_not:N \ttlh@runin }
3203   { \exp_not:N \ttlh@hang }
3204   \tl_tail:N #1
3205 }
3206 }
3207 \cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3208 {
3209   \cs_set_nopar:Npx #1 ##1
3210   {
3211     \bool_if:cTF { CTEX@#2@runin }
3212     { \exp_not:N \ttlh@runin }
3213     { \exp_not:N \ttlh@hang }
3214     \exp_args:No \tl_tail:n { #1 { } }
3215   }
3216 }
3217 \cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
3218 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3219 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3220 {
3221   \exp_not:n { {#1} {#2} {#3} {#4} }
3222   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3223 }
3224 \@ifpackageloaded { titlesec }
3225 { }
3226 {
3227   \ctex_at_end_package:nn { titlesec }
3228   {
3229     \@ifpackagewith { titlesec } { loadonly }
3230     { }
3231     { \ctex_titlesec_hook: }
3232   }
3233 }

```

让编译时终端显示 \CTEXthechapter, 目录使用 \CTEXtheXXX 编号。

```

3234 \ctex_at_end_package:nn { titlesec }
3235 {
3236   <*book|report>
3237   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3238   </book|report>
3239   \cs_if_free:NF \ttl@tocpart
3240   {
3241     \cs_set_protected:Npn \ttl@tocpart
3242     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3243   }
3244   \seq_map_inline:Nn \c__ctex_headings_seq
3245   {
3246     \cs_if_exist:cF { ttl@toc #1 }
3247     {
3248       \cs_new_protected:cpx { ttl@toc #1 }
3249       {
3250         \tl_set:Nn \exp_not:N \ttl@a
3251         {
3252           \exp_not:N \protect
3253           \exp_not:N \numberline { \exp_not:c { CTEXthe #1 } }
3254         }
3255       }
3256     }
3257   }
3258 }

```

14.8.2.8 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题¹⁹。

\ctex_titleps_hook: 我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```

3259 \group_begin:
3260 \char_set_catcode_other:N \#
3261 \cs_new_protected:Npn \ctex_titleps_hook:
3262 {
3263   \ctex_patch_cmd:Nnn \ttl@settopmark
3264   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3265   {
3266     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3267     \CTEX@titlepslabel@set {#1}
3268   }
3269   \ctex_patch_cmd:Nnn \ttl@setsubmark
3270   { \protect \@namedef { the#1 } { } }
3271   {
3272     \protect \@namedef { the#1 } { }
3273     \CTEX@titlepslabel@clear {#1}
3274   }
3275   \ctex_patch_cmd:Nnn \ttl@setsubmark
3276   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3277   {
3278     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3279     \CTEX@titlepslabel@set {#2}
3280   }
3281 }
3282 \group_end:

```

\CTEX@titlepslabel@set
\CTEX@titlepslabel@clear 这两个函数要在随后被 \xdef 展开来获得 \CTEXtheXXX 的内容, 不应该用 \protected 来定义。

```

3283 \cs_new:Npn \CTEX@titlepslabel@set #1
3284 {
3285   \cs_if_free:cF { CTEXthe#1 }
3286   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3287 }
3288 \cs_new:Npn \CTEX@titlepslabel@clear #1
3289 {
3290   \cs_if_free:cF { CTEXthe#1 }
3291   { \protect \@namedef { CTEXthe#1 } { } }
3292 }

```

titleps 宏包的功能可以由 titlesec 的选项 pagestyles 引入。

```

3293 \ctex_at_end_package:nn { titlesec }
3294 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3295 \ctex_at_end_package:nn { titleps } { \ctex_titleps_hook: }

```

除此之外, 也可以使用 titleps 提供的命令 \newtitlemark 来完成:

```

\newtitlemark { \CTEXthechapter }
\newtitlemark { \CTEXthesection }

```

但 \newtitlemark 不包含章节间的层次信息, 功能上不及修改内部命令完整。

\ttl@setifthe 使 \iftheXXX 等命令在页眉设置中可用。

```

3296 \ctex_at_end_package:nn { titleps }
3297 {

```

¹⁹<https://github.com/CTeX-org/ctex-kit/issues/217>

```

3298 \cs_set_protected:Npn \ttl@setifthe #1
3299 {
3300   \exp_args:Nco \cs_set:Npn { ifthe #1 }
3301   {
3302     \CTEXifname
3303     { \protect \@firstoftwo }
3304     { \protect \@secondoftwo }
3305   }
3306 }
3307 \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3308 }

```

14.8.3 目录标签的宽度

```

3309 \cs_new_protected:Npn \CTEX@toc@width@n #1
3310 {
3311   \hbox_set:Nn \l__ctex_tmp_box {#1}
3312   \dim_set:Nn \@tempdima
3313   {
3314     \dim_max:nn { \@tempdima }
3315     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3316   }
3317 }

```

\numberline

为 \numberline 命令打补丁,并兼容 tocloft 和 titletoc 宏包。

__ctex_patch_toc_width:n

这里需要替换 # 本身,因此需要先切换为 other 类。表示参数的 # 用 \c_parameter_token 代替。

```

3318 \group_begin:
3319 \char_set_catcode_other:N \#
3320 \use:n
3321 {
3322   \group_end:
3323   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3324   { \CTEX@toc@width@n {#1} }
3325   { }
3326   { \ctex_patch_failure:N \numberline }
3327   \cs_new_protected:Npn \__ctex_patch_toc_width:n \c_parameter_token 1
3328   {
3329     \@ifpackageloaded { \c_parameter_token 1 }
3330     { }
3331     {
3332       \ctex_at_end_package:nn { \c_parameter_token 1 }
3333       {
3334         \ctex_preto_cmd:NnnTF \numberline
3335         { \char_set_catcode_letter:n { 64 } }
3336         { \CTEX@toc@width@n {#1} }
3337         { }
3338         { \ctex_patch_failure:N \numberline }
3339       }
3340     }
3341   }
3342 }
3343 \__ctex_patch_toc_width:n { tocloft }
3344 \__ctex_patch_toc_width:n { titletoc }

```

14.8.4 页眉信息的修改

```

3345 <*article>
3346 \if@twoside
3347   \ctex_patch_cmd:Nnn \ps@headings
3348   { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3349   { \CTEXifname { \CTEXthesection \quad } { } }

```

```

3350 \ctex_patch_cmd:Nnn \ps@headings
3351 { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3352 { \CTEXifname { \CTEXthesubsection \quad } { } }
3353 \else:

```

不知为何,标准文档类此处对 `secnumdepth` 的判断为 0,与 `\section` 的层次 1 不符。

```

3354 \ctex_patch_cmd:Nnn \ps@headings
3355 { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3356 { \CTEXifname { \CTEXthesection \quad } { } }
3357 \fi:
3358 </article>
3359 <*book|report>
3360 \ctex_patch_cmd:Nnn \ps@headings
3361 {
3362   \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3363   \ifnum \c@secnumdepth > \m@ne
3364     \@chapapp \thechapter . ~ \ %
3365   \fi
3366   \fi \fi
3367 }
3368 { \CTEXifname { \CTEXthechapter \quad } { } }
3369 \if@twoside
3370 \ctex_patch_cmd:Nnn \ps@headings
3371 { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3372 { \CTEXifname { \CTEXthesection \quad } { } }
3373 \fi:
3374 </book|report>

```

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

```

3375 \ctex_at_end_package:nn { fancyhdr }
3376 {
3377   \ctex_patch_cmd:Nnn \f@nch@initialise
3378   { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3379   { \CTEXifname { \CTEXthesection \quad } { } }
3380   \ctex_patch_cmd:Nnn \f@nch@initialise
3381   { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3382   { \CTEXifname { \CTEXthesubsection \quad } { } }
3383   \ctex_patch_cmd:Nnn \f@nch@initialise
3384   { \ifnum \c@secnumdepth > \m@ne \@chapapp\thechapter . ~ \ \fi }
3385   { \CTEXifname { \CTEXthechapter \quad } { } }
3386   \ctex_patch_cmd:Nnn \f@nch@initialise
3387   { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3388   { \CTEXifname { \CTEXthesection \quad } { } }
3389   \f@nch@initialise

```

`fancyhdr` 的 `headings` 选项会重定义 `\ps@headings`,这里也要打补丁。

```

3390 \@ifpackagewith { fancyhdr } { headings }
3391 {
3392 <*article>
3393   \if@twoside
3394     \ctex_patch_cmd:Nnn \ps@headings
3395     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3396     { \CTEXifname { \CTEXthesection \quad } { } }
3397     \ctex_patch_cmd:Nnn \ps@headings
3398     { \ifnum \c@secnumdepth > \@ne \thesubsection \quad \fi }
3399     { \CTEXifname { \CTEXthesubsection \quad } { } }
3400   \else:
3401     \ctex_patch_cmd:Nnn \ps@headings
3402     { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3403     { \CTEXifname { \CTEXthesection \quad } { } }
3404   \fi:
3405 </article>
3406 <*book|report>
3407   \ctex_patch_cmd:Nnn \ps@headings

```

```

3408      {
<book> 3409      \ifnum \c@secnumdepth > \m@ne \ifmainmatter
<report> 3410      \ifnum \c@secnumdepth > \m@ne
3411      \@chapapp \thechapter . ~ \ %
<report> 3412      \fi
<book> 3413      \fi \fi
3414      }
3415      { \CTEXifname { \CTEXthechapter \quad } { } }
3416      \if@twoside
3417      \ctex_patch_cmd:Nnn \ps@headings
3418      { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3419      { \CTEXifname { \CTEXthesection \quad } { } }
3420      \fi:
3421 </book|report>
3422      }
3423      { }
3424      }
3425 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

```

3426 <*beamer>
3427 \ExplSyntaxOff

```

对应 \partpage。

```

3428 \defbeamerfont*{part page}{CTEX}[1][1]{%
3429 \begin{group}
3430 % \centering
3431 % {\usebeamerfont{part name}%
3432 % \usebeamerfont{fg}[part name]\partname~\insertromanpartnumber}
3433 % \vskip1em\par
3434 \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3435 \CTEX@heading@format@initial
3436 \CTEX@part@format{%
3437 \CTEX@indentbox{\CTEX@part@indent}%
3438 \ifodd \CTEX@part@numbering
3439 \CTEX@partname \CTEX@part@aftername
3440 \fi
3441 \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3442 % \usebeamerfont{part title}\insertpart\par
3443 \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3444 \end{beamercolorbox}}%
3445 \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3446 \end{group}

```

对应 \sectionpage。

```

3447 \defbeamerfont*{section page}{CTEX}[1][1]{%
3448 \begin{group}
3449 % \centering
3450 % {\usebeamerfont{section name}%
3451 % \usebeamerfont{fg}[section name]\sectionname~\insertsectionnumber}
3452 % \vskip1em\par
3453 \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3454 \CTEX@heading@format@initial
3455 \CTEX@section@format{%
3456 \CTEX@indentbox{\CTEX@section@indent}%
3457 \ifodd \CTEX@section@numbering
3458 \CTEX@sectionname \CTEX@section@aftername
3459 \fi
3460 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3461 % \usebeamerfont{section title}\insertsection\par
3462 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3463 \end{beamercolorbox}}%
3464 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%

```

```
3465 \endgroup}
```

对应 \subsectionpage。

```
3466 \defbeamer*{subsection page}{CTEX}[1][1]{%
3467 \begin{group}
3468 % \centering
3469 % {\usebeamerfont{subsection name}%
3470 % \usebeamerfont{subsection name}\insertsubsectionnumber}
3471 % \vskip1em\par
3472 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3473 \CTEX@heading@format@initial
3474 \CTEX@subsection@format{%
3475 \CTEX@indentbox{\CTEX@subsection@indent}%
3476 \ifodd \CTEX@subsection@numbering
3477 \CTEX@subsectionname \CTEX@subsection@aftername
3478 \fi
3479 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3480 % \usebeamerfont{subsection title}\insertsubsection\par
3481 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3482 \end{beamercolorbox}}%
3483 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3484 \endgroup}
```

将 beamer 的默认模板重定向为 CTEX 模板。

```
3485 \defbeamer*{part page}{default}{CTEX}
3486 \defbeamer*{section page}{default}{CTEX}
3487 \defbeamer*{subsection page}{default}{CTEX}
3488 \ExplSyntaxOn
3489 </beamer>
```

14.8.6 标题编号和目录的层次设置

secnumdepth **tocdepth** secnumdepth 在 beamer 下无意义。

```
3490 \ctex_define:n
3491 {
3492 <!*beamer>
3493   secnumdepth .code:n = \ctex_heading_depth:ne { secnumdepth } {#1} ,
3494   secnumdepth .value_required:n = true ,
3495 </!*beamer>
3496   tocdepth .code:n = \ctex_heading_depth:ne { tocdepth } {#1} ,
3497   tocdepth .value_required:n = true
3498 }
```

\ctex_heading_depth:nn 注意此处 \setcounter 的赋值是全局的。

```
3499 \cs_new_protected:Npn \ctex_heading_depth:nn #1#2
3500 {
3501   \prop_get:NnNTF \c__ctex_heading_level_prop {#2} \l__ctex_tmp_tl
3502   { \setcounter {#1} { \l__ctex_tmp_tl } }
3503   { \setcounter {#1} { \int_eval:n {#2} } }
3504 }
3505 \cs_generate_variant:Nn \ctex_heading_depth:nn { ne }
```

\c__ctex_heading_level_prop 章节层次与名称的对应表。

```
3506 \prop_const_from_keyval:Nn \c__ctex_heading_level_prop
3507 {
3508 <*article|beamer>
3509   part = 0 ,
3510 </article|beamer>
3511 <*book|report>
3512   part = -1 ,
3513   chapter = 0 ,
3514 </book|report>
```



```

3515     section      = 1 ,
3516     subsection   = 2 ,
3517     subsubsection = 3 ,
3518     paragraph     = 4 ,
3519     subparagraph  = 5
3520 }

```

14.8.7 标签引用数字的汉化

`\refstepcounter` 对标题进行引用时,设置标签为通过 `number` 选项设置的形式。

```

3521 \cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
3522 {
3523   \protected@edef \@currentlabel
3524   {
3525     \cs_if_exist:cTF { CTEX@the#1 }
3526     { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3527     { \exp_not:o { \@currentlabel } }
3528   }
3529 }

```

`\ctex_varioref_hook:` 关于标签引用的宏包可能会修改 `\refstepcounter`。其中 `cleveref` 和 `hyperref` 宏包都会保存之前的定义,并且它们都要求尽可能晚的被载入,所以对我们上述的修改影响不大。需要注意的是 `varioref` 宏包,如果它在 `CTEX` 之后被载入,我们之前的修改将会被覆盖。较新版 `LATEX` 内核已经包含 `\labelformat`,可以直接使用。

```

3530 \cs_new_protected:Npn \ctex_varioref_hook:
3531 {
3532   \seq_map_inline:Nn \c__ctex_headings_seq
3533   { \ctex_fix_varioref_label:n { ##1 } }
3534   \ctex_at_end_package:nn { cleveref } { \ctex_cleveref_hook: }
3535 }

```

`\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。

```

3536 \cs_new_protected:Npn \ctex_fix_varioref_label:n #1
3537 {
3538   \tl_if_empty:cT { p@#1 }
3539   { \exp_args:Nnx \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
3540 }

```

`\ctex_cleveref_hook:` 需要将 `cleveref` 包对应命令中 `\p@<counter>` 的参数及时展开,以兼容 `\labelformat`。
`__ctex_cleveref_hook_aux:N`

```

3541 \cs_new_protected:Npn \ctex_cleveref_hook:
3542 {
3543   \ifpackageloaded { hyperref }
3544   { \__ctex_cleveref_hook_aux:N \H@refstepcounter }
3545   {
3546     \__ctex_cleveref_hook_aux:N \refstepcounter@noarg
3547     \__ctex_cleveref_hook_aux:N \refstepcounter@optarg
3548   }
3549   \__ctex_cleveref_hook_aux:N \appendix
3550 }
3551 \cs_new_protected:Npn \__ctex_cleveref_hook_aux:N #1
3552 {
3553   \ctex_patch_cmd_all:NnnnTF #1

```

```

3554     {
3555         \ExplSyntaxOff
3556         \char_set_catcode_letter:n { 64 }
3557     }
3558     { \endcsname \csname the }
3559     { \expandafter \endcsname \csname the }
3560     { }
3561     { \ctex_patch_failure:N #1 }
3562 }

```

如果 `varioref` 已经被载入,则使用它来设置。

```

3563 \cs_if_exist:NTF \labelformat
3564 { \ctex_varioref_hook: }
3565 {
3566     \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3567     \RenewDocumentCommand \refstepcounter { m }
3568     {
3569         \CTEX@save@refstepcounter {#1}
3570         \CTEX@setcurrentlabel@n {#1}
3571     }
3572     \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3573 }

```

14.8.8 载入 `<scheme>` 文件

```

3574 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3575 </class|heading>

```

14.8.9 标题格式的 `scheme` 定义

下面使用 C_TE_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。

```

3576 <*scheme&(article|book|report|beamer)>

3577 \ctex_set:nn { part }
3578 {
3579     aftertitle = \par ,
3580 <*article|book|report>
3581     hang       = false ,
3582 </article|book|report>
3583 <*plain>
3584     name       = \partname \space ,
3585 <*article|book|report>
3586     number    = \thepart ,
3587 </article|book|report>
3588 <*beamer>
3589     number    = \insertromanpartnumber ,
3590 </beamer>
3591 </plain>
3592 <*chinese>
3593     number    = \chinese { part } ,
3594 </chinese>
3595 <*article>
3596     beforeskip = 4ex ,
3597     afterskip  = 3ex ,
3598 <*plain>
3599     format     = \raggedright ,
3600     nameformat = \Large \bfseries ,
3601     aftername  = \par \nobreak ,
3602     titleformat = \huge \bfseries ,
3603     afterindent = false
3604 </plain>
3605 <*chinese>

```

```

3606     format      = \Large \bfseries \centering ,
3607     aftername    = \quad ,
3608     afterindent  = true
3609 </chinese>
3610 </article>
3611 <*book|report>
3612     aftername    = \par \vskip 20 \p@ ,
3613     before skip  = 0pt \@plus 1fil ,
3614     after skip   = 0pt \@plus 1fil ,
3615     page style   = plain ,
3616     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3617 <*plain>
3618     format      = \centering ,
3619     name format  = \huge \bfseries ,
3620     title format = \Huge \bfseries
3621 </plain>
3622 <*chinese>
3623     format      = \huge \bfseries \centering
3624 </chinese>
3625 </book|report>
3626 <*beamer>
3627     format      = \centering ,
3628     name format  = \usebeamerfont { part ~ name }
3629                 \usebeamercolor [fg] { part ~ name } ,
3630     aftername    = \vskip 1em \par ,
3631     title format = \usebeamerfont { part ~ title }
3632 </beamer>
3633 }

3634 <*book|report>
3635 \ctex_set:nn { chapter }
3636 {
3637     page style   = plain ,
3638     after title  = \par ,
3639     hang        = false ,
3640     before skip  = 50 \p@ ,
3641     after skip   = 40 \p@ ,
3642     lof skip     = 10 \p@ ,
3643     lot skip     = 10 \p@ ,
3644     break       = \if@openright \cleardoublepage \else \clearpage \fi ,
3645 <*plain>
3646     name        = \chaptername \space ,
3647     number      = \thechapter ,
3648     format      = \raggedright ,
3649     name format  = \huge \bfseries ,
3650     aftername    = \par \nobreak \vskip 20 \p@ ,
3651     title format = \Huge \bfseries ,
3652     afterindent  = false ,
3653     tocline     = \CTEXnumberline {#1} #2
3654 </plain>
3655 <*chinese>
3656     number      = \chinese { chapter } ,
3657     format      = \huge \bfseries \centering ,
3658     aftername    = \quad ,
3659     afterindent  = true
3660 </chinese>
3661 }
3662 </book|report>

3663 \ctex_set:nn { section }
3664 {
3665 <*article|book|report>
3666     number      = \thesection ,
3667     aftername    = \quad ,
3668     after title  = \@@par ,
3669     before skip  = 3.5ex \@plus 1ex \@minus .2ex ,
3670     after skip   = 2.3ex \@plus .2ex ,
3671     runin       = false ,

```

```

3672     break          = \addpenalty \@secpenalty ,
3673 <*plain>
3674     format          = \Large \bfseries ,
3675     afterindent     = false
3676 </plain>
3677 <*chinese>
3678     format          = \Large \bfseries \centering ,
3679     afterindent     = true
3680 </chinese>
3681 </article|book|report>
3682 <*beamer>
3683 <*plain>
3684     name            = \sectionname \space ,
3685 </plain>
3686     format          = \centering ,
3687     number          = \insertsectionnumber ,
3688     nameformat      = \usebeamerfont { section ~ name }
3689                     \usebeamercolor [fg] { section ~ name } ,
3690     aftername       = \vskip 1em \par ,
3691     titleformat     = \usebeamerfont { section ~ title } ,
3692     aftertitle      = \par
3693 </beamer>
3694 }

3695 \ctex_set:nn { subsection }
3696 {
3697 <*article|book|report>
3698     number          = \thesubsection ,
3699     format          = \large \bfseries ,
3700     aftername       = \quad ,
3701     aftertitle      = \@@par ,
3702     beforekip       = 3.25ex \@plus 1ex \@minus .2ex ,
3703     afterskip       = 1.5ex \@plus .2ex ,
3704     runin           = false ,
3705     break           = \addpenalty \@secpenalty ,
3706 <*plain>
3707     afterindent     = false
3708 </plain>
3709 <*chinese>
3710     afterindent     = true
3711 </chinese>
3712 </article|book|report>
3713 <*beamer>
3714 <*plain>
3715     name            = \subsectionname \space ,
3716     number          = \insertsubsectionnumber ,
3717 </plain>
3718 <*chinese>
3719     number          = \arabic { section } . \arabic { subsection } ,
3720 </chinese>
3721     format          = \centering ,
3722     nameformat      = \usebeamerfont { subsection ~ name }
3723                     \usebeamercolor [fg] { subsection ~ name } ,
3724     aftername       = \vskip 1em \par ,
3725     titleformat     = \usebeamerfont { subsection ~ title } ,
3726     aftertitle      = \par
3727 </beamer>
3728 }

3729 <*article|book|report>

3730 \ctex_set:nn { subsubsection }
3731 {
3732     number          = \thesubsubsection ,
3733     format          = \normalsize \bfseries ,
3734     aftername       = \quad ,
3735     aftertitle      = \@@par ,
3736     beforekip       = 3.25ex \@plus 1ex \@minus .2ex ,

```

```

3737     afterskip    = 1.5ex \@plus .2ex ,
3738     runin        = false ,
3739     break        = \addpenalty \@secpenalty ,
3740 <*plain>
3741     afterindent = false
3742 </plain>
3743 <*chinese>
3744     afterindent = true
3745 </chinese>
3746 }

3747 \ctex_set:nn { paragraph }
3748 {
3749     number        = \theparagraph ,
3750     format        = \normalsize \bfseries ,
3751     aftername     = \quad ,
3752     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3753     break        = \addpenalty \@secpenalty ,
3754 <*plain>
3755     afterindent = false
3756 </plain>
3757 <*chinese>
3758     afterindent = true
3759 </chinese>
3760 }

3761 \ctex_set:nn { subparagraph }
3762 {
3763     number        = \thesubparagraph ,
3764     format        = \normalsize \bfseries ,
3765     aftername     = \quad ,
3766     beforekip     = 3.25ex \@plus 1ex \@minus .2ex ,
3767     break        = \addpenalty \@secpenalty ,
3768 <*plain>
3769     afterindent = false
3770 </plain>
3771 <*chinese>
3772     afterindent = true
3773 </chinese>
3774 }

```

处理 sub3section 与 sub4section 的格式。

```

3775 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3776 {
3777     \ctex_set:nn { paragraph }
3778     {
3779         aftertitle = \@@par ,
3780         afterskip  = 1ex \@plus .2ex ,
3781         runin      = false
3782     }
3783 }
3784 {
3785     \ctex_set:nn { paragraph }
3786     {
3787         afterskip  = 1em ,
3788         runin      = true
3789     }
3790 }
3791 \int_compare:nNnTF \g__ctex_section_depth_int > 3
3792 {
3793     \ctex_set:nn { subparagraph }
3794     {
3795         aftertitle = \@@par ,
3796         afterskip  = 1ex \@plus .2ex ,
3797         runin      = false
3798     }
3799 }

```

```

3800 {
3801   \ctex_set:nn { subparagraph }
3802   {
3803     afterskip = 1em ,
3804     runin     = true
3805   }
3806 }
3807 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3808 { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
3809 { \ctex_set:nn { subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

3810 \ctex_set:nn { appendix }
3811 <article>
3812 { number = \@Alph \c@section }
3813 </article>
3814 <book|report>
3815 {
3816   name = \appendixname \space ,
3817   number = \@Alph \c@chapter
3818 }
3819 </book|report>
3820 </article|book|report>
3821 </scheme&(article|book|report|beamer)>

```

14.8.10 ctex.sty 的 heading 选项

```

3822 <ctex|ctexheading>

```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3823 \clist_map_inline:nn { article , book , report , beamer }
3824 {
3825   \@ifclassloaded {#1}
3826   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3827   { }
3828 }

```

若标准文档类被引入,则载入对应的标题定义文件。否则视 \chapter 是否有定义来引入 book 或者 article。

```

3829 \msg_new:nnn { ctex } { not-standard-class }
3830 {
3831   None~of~the~standard~document~classes~was~loaded.\\
3832   Heading~`#1'~is~selected.\\
3833   ctex~may~not~work~as~expected.
3834 }
3835 <ctex> \bool_if:NTF \l__ctex_heading_bool
3836 <ctexheading> \use:n
3837 {
3838   \tl_if_exist:NTF \c__ctex_std_class_tl
3839   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3840   {
3841     \cs_if_exist:NTF \chapter
3842     {
3843       \cs_if_exist:NF \if@mainmatter
3844       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3845       \tl_const:Nn \c__ctex_class_tl { book }
3846     }
3847     { \tl_const:Nn \c__ctex_class_tl { article } }
3848     \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3849   }
3850   \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }
3851 }
3852 <ctex> { \ctex_scheme_input:o { \l__ctex_scheme_tl } }

```

3853 `</ctex|ctexheading>`

14.8.11 标题配置文件

```

3854 <*name>
3855 \keys_set_known:nn { ctex }
3856 {
3857   contentsname = 目录 ,
3858   listfigurename = 插图 ,
3859   listtablename = 表格 ,
3860   figurename = 图 ,
3861   tablename = 表 ,
3862   abstractname = 摘要 ,
3863   indexname = 索引 ,
3864   bibname = 参考文献 ,
3865   appendixname = 附录 ,
3866   proofname = 证明 ,
3867   algorithmname = 算法 ,
3868   refname = 参考文献 ,
3869   continuation = (续) ,
3870   part / name = { 第 , 部分 } ,
3871   chapter / name = { 第 , 章 }
3872 }
3873 </name>

```

14.9 chinese 方案的其他设置

3874 `<*scheme&chinese>`

`chinese` 在标准文档类下的页面格式总采用 `headings`。

`<article|book|report>` 3875 `\pagestyle { headings }`

日期格式。

3876 `\ctex_set:n { today = small }`

若用户未设置宏包选项 `autoindent`, 则自动调整首行缩进。

```

3877 \ctex_if_autoindent_touched:F
3878 { \ctex_set:n { autoindent = true } }

```

使用标题定义时的设置。首先是命题名字汉化。`beamer` 需要汉化定理名称。

```

3879 <*!generic>
3880 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3881 <*beamer>
3882 {
3883   \uselanguage { ChineseGBK }
3884   \languagealias { chinese } { ChineseGBK }
3885   \ctex_file_input:n { ctex-name-gbk.cfg }
3886 }
3887 {
3888   \uselanguage { ChineseUTF8 }
3889   \languagealias { chinese } { ChineseUTF8 }
3890   \ctex_file_input:n { ctex-name-utf8.cfg }
3891 }

```

让 `translator` 包优先查找中文翻译。

```

3892 \clist_put_left:Nn \trans@languagepath { chinese }
3893 </beamer>
3894 <*!beamer>
3895 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3896 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

对 `beamer` 以外的文档类, 若用户未设置宏包选项 `zihao`, 则设置 `\normalsize` 为五号字。`beamer` 不调整默认字体大小。

```

3897 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3898 { \int_gset:Nn \g__ctex_font_size_int { 0 } }

```

对 **beamer** 以外的文档类, 若用户未设置宏包选项 `linespread`, 则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。beamer 不调整行距。

```

3899 \fp_if_nan:nT { \l__ctex_line_spread_fp }
3900 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3901 </!beamer>
3902 </!generic>

```

不使用标题定义时的通用设置, 注意此处 `\c__ctex_std_class_tl` 可能没有定义。

```

3903 <*generic>
3904 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3905 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3906 {
3907   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3908   {
3909     \uselanguage { ChineseGBK }
3910     \languagealias { chinese } { ChineseGBK }
3911     \ctex_file_input:n { ctex-name-gbk.cfg }
3912   }
3913   {
3914     \uselanguage { ChineseUTF8 }
3915     \languagealias { chinese } { ChineseUTF8 }
3916     \ctex_file_input:n { ctex-name-utf8.cfg }
3917   }
3918   \clist_put_left:Nn \trans@languagepath { chinese }
3919 }
3920 {
3921   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3922   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3923   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3924   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3925   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
3926   \fp_if_nan:nT { \l__ctex_line_spread_fp }
3927   { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }

```

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

```

3928   \tl_if_exist:NT \c__ctex_std_class_tl
3929   { \RequirePackage { indentfirst } }
3930 }
3931 </generic>
3932 </scheme&chinese>

```

14.10 中文字号

```

3933 <*class|ctex|ctexsize>
3934 \NewDocumentCommand \zihao { m }
3935 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

3936 \cs_new_protected:Npn \ctex_zihao:n #1
3937 {
3938   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3939   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3940   { \msg_error:nnn { ctex } { fontsize } {#1} }
3941 }
3942 \msg_new:nnnn { ctex } { fontsize }
3943 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3944 {
3945   The~old~font~size~is~used~if~you~continue.\\
3946   The~available~font~sizes~are~listed~as~follow.\\
3947   \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3948 }

```


14.10.1 定义中文字号

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²⁰。

```

\c__ctex_font_size_prop
\__ctex_save_font_size:nn

3949 \prop_new:N \c__ctex_font_size_prop
3950 \seq_new:N \c__ctex_font_size_seq
3951 \cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
3952 {
3953   \use:x
3954   {
3955     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3956     {
3957       { \dim_to_decimal:n {#2} }
3958       { \dim_to_decimal:n { (#2) * 6 / 5 } }
3959     }
3960   }
3961   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3962 }
3963 \clist_map_inline:nn
3964 {
3965   { 8 } { 5 bp } ,
3966   { 7 } { 5.5 bp } ,
3967   { -6 } { 6.5 bp } ,
3968   { 6 } { 7.5 bp } ,
3969   { -5 } { 9 bp } ,
3970   { 5 } { 10.5 bp } ,
3971   { -4 } { 12 bp } ,
3972   { 4 } { 14 bp } ,
3973   { -3 } { 15 bp } ,
3974   { 3 } { 16 bp } ,
3975   { -2 } { 18 bp } ,
3976   { 2 } { 22 bp } ,
3977   { -1 } { 24 bp } ,
3978   { 1 } { 26 bp } ,
3979   { -0 } { 36 bp } ,
3980   { 0 } { 42 bp }
3981 }
3982 { \__ctex_save_font_size:nn #1 }

3983 \cs_new_protected:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3984 {
3985   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3986   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3987 }

3988 \cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
3989 {
3990   \tl_clear:N #1
3991   \tl_map_inline:nn {#2}
3992   {
3993     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3994     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3995     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3996   }
3997 }

3998 \clist_map_inline:nn
3999 {
4000   { 8 } { 8 } { 5pt } { 5pt } ,
4001   { 7 } { 7 } { 5pt } { 5pt } ,
4002   { -6 } { -6 } { 5pt } { 5pt } ,
4003   { 6 } { 6 } { 5pt } { 5pt } ,
4004   { -5 } { -5 } { 6pt } { 5pt } ,
4005   { 5 } { 5 } { 7pt } { 5pt } ,

```

²⁰<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

4006 { -4 }{ -4 }{ 8pt }{ 6pt } ,
4007 { 4 }{ 4 }{ 5 }{ 6 } ,
4008 { -3 }{ -3 }{ -4 }{ -5 } ,
4009 { 3 }{ 3 }{ 4 }{ 5 } ,
4010 { -2 }{ -2 }{ -3 }{ -4 } ,
4011 { 2 }{ 2 }{ 3 }{ 4 } ,
4012 { -1 }{ -1 }{ -2 }{ -3 } ,
4013 { 1 }{ 1 }{ 2 }{ 3 } ,
4014 { -0 }{ -0 }{ -1 }{ -2 } ,
4015 { 0 }{ 0 }{ 1 }{ 2 }
4016 }
4017 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

4018 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
4019 {
4020   \prop_get:NnNTF \c_ctex_font_size_prop {#2} \l__ctex_font_size_tl
4021   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
4022   { \msg_error:nnn { ctex } { fontsize } {#2} }
4023 }
4024 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
4025 { \cs_set_protected:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

4026 \int_case:nn { \g__ctex_font_size_int }
4027 {
4028   { 0 } { \ctex_file_input:n { ctex-c5size.clo } }
4029   { 1 } { \ctex_file_input:n { ctex-cs4size.clo } }
4030 }
4031 </class>ctex<ctexsize>
4032 <*c5size>
4033 \ctex_set_font_size:Nnn \normalsize { 5 }
4034 {
4035   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
4036   \abovedisplayshortskip \z@ \@plus3\p@
4037   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
4038   \belowdisplayskip \abovedisplayskip
4039   \let\@listi\@listI
4040 }
4041 \ctex_set_font_size:Nnn \small { -5 }
4042 {
4043   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
4044   \abovedisplayshortskip \z@ \@plus2\p@
4045   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
4046   \def\@listi{\leftmargin\leftmarginI
4047             \topsep 4\p@ \@plus2\p@ \@minus2\p@
4048             \parsep 2\p@ \@plus\p@ \@minus\p@
4049             \itemsep \parsep}
4050   \belowdisplayskip \abovedisplayskip
4051 }
4052 \ctex_set_font_size:Nnn \footnotesize { 6 }
4053 {
4054   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
4055   \abovedisplayshortskip \z@ \@plus\p@
4056   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
4057   \def\@listi{\leftmargin\leftmarginI
4058             \topsep 3\p@ \@plus\p@ \@minus\p@
4059             \parsep 2\p@ \@plus\p@ \@minus\p@
4060             \itemsep \parsep}
4061   \belowdisplayskip \abovedisplayskip
4062 }
4063 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
4064 \ctex_set_font_size:Nnn \tiny { 7 } { }
4065 \ctex_set_font_size:Nnn \large { -4 } { }
4066 \ctex_set_font_size:Nnn \Large { -3 } { }

```

```

4067 \ctex_set_font_size:Nnn \LARGE { -2 } { }
4068 \ctex_set_font_size:Nnn \huge { 2 } { }
4069 \ctex_set_font_size:Nnn \Huge { 1 } { }
4070 </cs5size>
4071 <*cs4size>
4072 \ctex_set_font_size:Nnn \normalsize { -4 }
4073 {
4074   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
4075   \abovedisplayshortskip \z@ \@plus3\p@
4076   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
4077   \belowdisplayskip \abovedisplayskip
4078   \let\@listi\@listI
4079 }
4080 \ctex_set_font_size:Nnn \small { 5 }
4081 {
4082   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
4083   \abovedisplayshortskip \z@ \@plus3\p@
4084   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
4085   \def\@listi{\leftmargin\leftmarginI
4086             \topsep 9\p@ \@plus3\p@ \@minus5\p@
4087             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
4088             \itemsep \parsep}
4089   \belowdisplayskip \abovedisplayskip
4090 }
4091 \ctex_set_font_size:Nnn \footnotesize { -5 }
4092 {
4093   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
4094   \abovedisplayshortskip \z@ \@plus3\p@
4095   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
4096   \def\@listi{\leftmargin\leftmarginI
4097             \topsep 6\p@ \@plus2\p@ \@minus2\p@
4098             \parsep 3\p@ \@plus2\p@ \@minus\p@
4099             \itemsep \parsep}
4100   \belowdisplayskip \abovedisplayskip
4101 }
4102 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
4103 \ctex_set_font_size:Nnn \tiny { -6 } { }
4104 \ctex_set_font_size:Nnn \large { -3 } { }
4105 \ctex_set_font_size:Nnn \Large { -2 } { }
4106 \ctex_set_font_size:Nnn \LARGE { 2 } { }
4107 \ctex_set_font_size:Nnn \huge { -1 } { }
4108 \ctex_set_font_size:Nnn \Huge { 1 } { }
4109 </cs4size>
<ctexsize> 4110 \normalsize
4111 <*class|ctex>

```

14.11 更新行距

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

```

4112 \fp_if_nan:nF { \l__ctex_line_spread_fp }
4113 {
4114   \exp_args:Nx \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。

```

4115   \group_begin: \footnotesize \exp_args:NNNo \group_end:
4116   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
4117 }

```

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

```

4118 \normalsize

```

14.12 其他功能

`\CTeX` `ctex-faq.sty` 中的定义是

```
\DeclareRobustCommand\CTeX{${\mathbb{C}}$\kern-.05em\TeX}
```

然而 `\mathbb` 未必有定义, 这里就不采用它了, 只定义最简单的形式。`CTeX` 可以直接用在 PDF 书签中。

```
4119 \NewDocumentCommand \CTeX { }
4120 { C \TeX }
4121 \ctex_at_end_package:nn { hyperref }
4122 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

`captiondelimenter` 过时选项。

```
4123 \ctex_define:n
4124 {
4125     captiondelimenter .code:n =
4126     {
4127         \ctex_deprecated_option:n
4128         { You~can~load~the~package~`caption'~to~get~its~functionality. }
4129     }
4130 }
4131 </class|ctex>
```

14.12.1 列表环境的缩进

`\verse`
`\quotation` 只在使用文档类的时候修改诗歌和引用环境的缩进。

```
4132 <*scheme&chinese&(article|book|report)>
4133 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
4134 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
4135 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
4136 </scheme&chinese&(article|book|report)>
4137 <*class|ctex>
```

14.13 载入中文字库

`\ctex_fontset_error:n` 字库不可用时给出紧急错误信息, 停止读取定义文件。

```
4138 \cs_new_protected:Npn \ctex_fontset_error:n #1
4139 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
4140 \msg_new:nnn { ctex } { fontset-unavailable }
4141 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }
```

`\ctex_fontset_case:nnn` 3 个参数依次为 `pdfTeX`、`upTeX` 和 `XYTeX/LuaTeX`。

```
4142 \cs_new:Npx \ctex_fontset_case:nnn #1#2#3
4143 {
4144     \sys_if_engine_pdftex:TF
4145     {#1}
4146     { \sys_if_engine_uptex:TF {#2} {#3} }
4147 }
```

`\ctex_fontset_case:nnnn` 4 个参数依次为 `pdfTeX`(生成 PDF)、`pdfTeX`(生成 DVI)、`upTeX` 和 `XYTeX/LuaTeX`。

```
4148 \cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
4149 {
4150     \sys_if_engine_pdftex:TF
4151     { \sys_if_output_pdf:TF {#1} {#2} }
4152     { \sys_if_engine_uptex:TF {#3} {#4} }
4153 }
```

`\ctex_detect_platform:` 根据操作系统判断默认字体配置。

```

4154 \cs_new_protected:Npn \ctex_detect_platform:
4155 {
4156   \sys_if_platform_windows:TF
4157   { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
4158   {
4159     \ctex_if_platform_macos:TF
4160     { \tl_gset:Nn \g__ctex_fontset_tl { mac } }
4161     { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
4162   }
4163 }

```

`\ctex_if_platform_macos:TF` 以特定字体判断 macOS 系统。

```

4164 \cs_new_protected:Npn \ctex_if_platform_macos:TF
4165 { \file_if_exist:nTF { /System/Library/Fonts/Menlo.ttc } }

```

`\ctex_load_fontset:` 如果用户没有指定字体,则探测操作系统,载入相应的字体配置。

```

4166 \cs_new_protected:Npn \ctex_load_fontset:
4167 {
4168   \tl_if_empty:NTF \g__ctex_fontset_tl
4169   { \ctex_detect_platform: }
4170   {
4171     \bool_lazy_or:nnTF
4172     { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
4173     { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
4174     {
4175       \msg_warning:nnxx { ctex } { deprecated-fontset }
4176       { \g__ctex_fontset_tl } { windows }
4177       \tl_gset:Nn \g__ctex_fontset_tl { windows }
4178     }
4179     {
4180       \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
4181       {
4182         \use:x
4183         {
4184           \ctex_detect_platform:
4185           \msg_error:nnxx { ctex } { fontset-not-found }
4186           { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
4187         }
4188       }
4189     }
4190   }
4191   \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
4192 }
4193 \msg_new:nnn { ctex } { deprecated-fontset }
4194 { CTeX~fontset~`#1'~is~deprecated.\\ Fontset~`#2'~will~be~used~instead. }
4195 \msg_new:nnnn { ctex } { fontset-not-found }
4196 { CTeX~fontset~`#1'~could~not~be~found.\\ Fontset~`#2'~will~be~used~instead. }
4197 { You~may~run~`mktexlsr'~firstly. }
4198 \@onlypreamble \ctex_load_fontset:

```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```

4199 \ctex_define:n
4200 {
4201   fontset .code:n =
4202   {
4203     \ctex_if_preamble:TF
4204     {
4205       \str_if_eq:eeTF {#1} { none }
4206       { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4207       {
4208         \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4209         {

```

```

4210             \tl_gset:Nx \g__ctex_fontset_tl {#1}
4211             \ctex_load_fontset:
4212         }
4213     {
4214         \msg_error:nnxx { ctex } { fontset-loaded }
4215         { \g__ctex_fontset_tl } {#1}
4216     }
4217 }
4218 }
4219 { \msg_error:nn { ctex } { fontset-only-preamble } }
4220 }
4221 }
4222 \msg_new:nnnn { ctex } { fontset-loaded }
4223 {
4224     CTEX~fontset~`#1'~has~been~loaded.
4225     \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
4226 }
4227 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4228 \msg_new:nnn { ctex } { fontset-only-preamble }
4229 { The~fontset'~option~can~be~used~only~in~preamble. }

```

载入中文字库。

```

4230 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4231 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

4232 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4233 </class>ctex>
4234 <*config>
4235 %%
4236 </config>

```

14.14.2 ctexopts.cfg

这里仅为配置文件示例:使用 Windows 下的字体设置。

```

4237 <*ctexopts>
4238 %%
4239 %% \ctex_set:nn { option } { fontset = windows }
4240 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

4241 <*c19|c70>
4242 %%
4243 %% Chinese characters
4244 %%
4245 <c19> %% character set: GBK (extension of GB 2312)
4246 <c70> %% character set: Unicode
4247 %% font encoding: Unicode
4248 %%
4249 </c19|c70>

```

CJK 宏包使用的字体族。

```

4250 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
4251 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}

```

```

<sf&c19> 4252 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 4253 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 4254 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 4255 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

upL^AT_EX 使用的字体族。upL^AT_EX 在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```

<rm&jy2> 4256 \DeclareKanjiFamily{JY2}{zhrm}{}
<rm&jt2> 4257 \DeclareKanjiFamily{JT2}{zhrm}{}
<sf&jy2> 4258 \DeclareKanjiFamily{JY2}{zhshf}{}
<sf&jt2> 4259 \DeclareKanjiFamily{JT2}{zhshf}{}
<tt&jy2> 4260 \DeclareKanjiFamily{JY2}{zhhtt}{}
<tt&jt2> 4261 \DeclareKanjiFamily{JT2}{zhhtt}{}

4262 <*rm>
4263 <*c19>
4264 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4265 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4266 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4267 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4268 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4269 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4270 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4271 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4272 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4273 </c19>
4274 <*c70>
4275 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4276 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4277 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4278 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4279 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4280 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4281 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4282 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4283 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4284 </c70>
4285 <*jy2>
4286 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{\CJKnormal}
4287 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{\CJKnormal}
4288 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{\CJKnormal}
4289 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{\CJKnormal}
4290 </jy2>
4291 <*jt2>
4292 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{\CJKnormal}
4293 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{\CJKnormal}
4294 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{\CJKnormal}
4295 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{\CJKnormal}
4296 </jt2>
4297 </rm>

4298 <*sf>
4299 <*c19>
4300 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4301 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4302 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4303 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4304 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4305 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4306 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4307 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4308 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4309 </c19>
4310 <*c70>
4311 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4312 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4313 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4314 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyou}{\CJKnormal}

```

```

4315 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyou1}{\CJKbold}
4316 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyou1}{\CJKbold}
4317 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4318 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4319 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4320 </c70>
4321 <*jy2>
4322 \DeclareFontShape{JY2}{zhfsf}{m}{n}{<-> upzhsans-h}{ }
4323 \DeclareFontShape{JY2}{zhfsf}{b}{n}{<-> upzhsansb-h}{ }
4324 \DeclareFontShape{JY2}{zhfsf}{bx}{n}{<-> upzhsansb-h}{ }
4325 </jy2>
4326 <*jt2>
4327 \DeclareFontShape{JT2}{zhfsf}{m}{n}{<-> upzhsans-v}{ }
4328 \DeclareFontShape{JT2}{zhfsf}{b}{n}{<-> upzhsansb-v}{ }
4329 \DeclareFontShape{JT2}{zhfsf}{bx}{n}{<-> upzhsansb-v}{ }
4330 </jt2>
4331 </sf>
4332 <*tt>
4333 <*c19>
4334 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4335 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4336 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4337 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4338 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4339 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4340 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4341 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4342 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4343 </c19>
4344 <*c70>
4345 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4346 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4347 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4348 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4349 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4350 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4351 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4352 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4353 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4354 </c70>
4355 <*jy2>
4356 \DeclareFontShape{JY2}{zhhtt}{m}{n}{<-> upzhmono-h}{ }
4357 </jy2>
4358 <*jt2>
4359 \DeclareFontShape{JT2}{zhhtt}{m}{n}{<-> upzhmono-v}{ }
4360 </jt2>
4361 </tt>

```

14.15.2 预定义字库

```
4362 <*fontset>
```

14.15.2.1 adobe

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

```

4363 <*adobe>
4364 \ctex_fontset_case:nnnn
4365 { \ctex_fontset_error:n { adobe } }
4366 {
4367   \ctex_zhmap_case:nnn
4368   {
4369     \setCJKmainfont { AdobeSongStd-Light.otf }
4370     [
4371       cmap          = UniGB-UTF16-H,
4372       BoldFont      = AdobeHeitiStd-Regular.otf,

```



```

4373         ItalicFont = AdobeKaitiStd-Regular.otf
4374     ]
4375     \setCJKsansfont { AdobeHeitiStd-Regular.otf }
4376     [ cmap = UniGB-UTF16-H ]
4377     \setCJKmonofont { AdobeFangsongStd-Regular.otf }
4378     [ cmap = UniGB-UTF16-H ]
4379     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf }
4380     [ cmap = UniGB-UTF16-H ]
4381     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular.otf }
4382     [ cmap = UniGB-UTF16-H ]
4383     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular.otf }
4384     [ cmap = UniGB-UTF16-H ]
4385     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular.otf }
4386     [ cmap = UniGB-UTF16-H ]
4387     \ctex_punct_set:n { adobe }
4388     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4389     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4390     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4391     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4392     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4393 }
4394 {
4395     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { adobe }
4396     \ctex_punct_set:n { adobe }
4397     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4398     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4399     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4400 }
4401 { \ctex_fontset_error:n { adobe } }
4402 }
4403 {
4404     \ctex_set_upfonts:nnnnnn
4405     { AdobeSongStd-Light.otf }
4406     { AdobeHeitiStd-Regular.otf }
4407     { AdobeKaitiStd-Regular.otf }
4408     { AdobeHeitiStd-Regular.otf }
4409     { AdobeHeitiStd-Regular.otf }
4410     { AdobeFangsongStd-Regular.otf }
4411     \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4412     \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4413     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4414     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4415 }
4416 {
4417     \setCJKmainfont { AdobeSongStd-Light }
4418     [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
4419     \setCJKsansfont { AdobeHeitiStd-Regular }
4420     \setCJKmonofont { AdobeFangsongStd-Regular }
4421     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4422     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4423     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4424     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4425 }
4426 </adobe>

```

14.15.2.2 fandol

```

4427 <fandol>
4428 \ctex_fontset_case:nnnn
4429 { \ctex_fontset_error:n { fandol } }
4430 {
4431     \ctex_zhmap_case:nnn
4432     {
4433         \setCJKmainfont { FandolSong-Regular.otf }
4434         [
4435             cmap = UniGB-UTF16-H,

```

```

4436         BoldFont = FandolSong-Bold.otf,
4437         ItalicFont = FandolKai-Regular.otf
4438     ]
4439     \setCJKsansfont { FandolHei-Regular.otf }
4440     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4441     \setCJKmonofont { FandolFang-Regular.otf }
4442     [ cmap = UniGB-UTF16-H ]
4443     \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
4444     [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
4445     \setCJKfamilyfont { zhhei } { FandolHei-Regular.otf }
4446     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4447     \setCJKfamilyfont { zhfs } { FandolFang-Regular.otf }
4448     [ cmap = UniGB-UTF16-H ]
4449     \setCJKfamilyfont { zhkai } { FandolKai-Regular.otf }
4450     [ cmap = UniGB-UTF16-H ]
4451     \ctex_punct_set:n { fandol }
4452     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4453     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4454     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4455     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4456     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4457     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4458 }
4459 {
4460     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { fandol }
4461     \ctex_punct_set:n { fandol }
4462     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4463     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4464     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4465 }
4466 { \ctex_fontset_error:n { fandol } }
4467 }
4468 {
4469     \ctex_set_upfonts:nnnnnn
4470     { FandolSong-Regular.otf }
4471     { FandolSong-Bold.otf }
4472     { FandolKai-Regular.otf }
4473     { FandolHei-Regular.otf }
4474     { FandolHei-Bold.otf }
4475     { FandolFang-Regular.otf }
4476     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4477     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4478     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4479     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4480 }
4481 {
4482     \setCJKmainfont { FandolSong-Regular }
4483     [
4484         Extension = .otf,
4485         BoldFont = FandolSong-Bold,
4486         ItalicFont = FandolKai-Regular
4487     ]
4488     \setCJKsansfont { FandolHei-Regular }
4489     [ Extension = .otf, BoldFont = FandolHei-Bold ]
4490     \setCJKmonofont { FandolFang-Regular }
4491     [ Extension = .otf ]
4492     \setCJKfamilyfont { zhsong } { FandolSong-Regular }
4493     [ Extension = .otf, BoldFont = FandolSong-Bold ]
4494     \setCJKfamilyfont { zhhei } { FandolHei-Regular }
4495     [ Extension = .otf, BoldFont = FandolHei-Bold ]
4496     \setCJKfamilyfont { zhfs } { FandolFang-Regular }
4497     [ Extension = .otf ]
4498     \setCJKfamilyfont { zhkai } { FandolKai-Regular }
4499     [ Extension = .otf ]
4500 }
4501 </fandol>

```

14.15.2.3 founder

```

4502 <*founder>
4503 \ctex_fontset_case:nnn
4504 {
4505   \ctex_zhmap_case:nnn
4506   {
4507     \setCJKmainfont { FZSSK.TTF }
4508     [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
4509     \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
4510     \setCJKmonofont { FZFSK.TTF }
4511     \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
4512     \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4513     \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4514     \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4515     \setCJKfamilyfont { zhli } { FZLSK.TTF }
4516     \setCJKfamilyfont { zhyou } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
4517     \ctex_punct_set:n { founder }
4518     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4519     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4520     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4521     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4522     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4523     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4524     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4525   }
4526   {
4527     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { founder }
4528     \ctex_punct_set:n { founder }
4529     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4530     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4531     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4532   }
4533   { \ctex_fontset_error:n { founder } }
4534 }
4535 {
4536   \ctex_set_upfonts:nnnnnn
4537   { FZSSK.TTF }
4538   { FZXBSK.TTF }
4539   { FZKTK.TTF }
4540   { FZXH1K.TTF }
4541   { FZHTK.TTF }
4542   { FZFSK.TTF }
4543   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4544   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4545   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4546   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4547   \ctex_set_upfamily:nnn { zhli } { upschrm } {}
4548   \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4549   \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4550   \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4551 }
4552 {
4553   \setCJKmainfont { FZShuSong-Z01 }
4554   [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
4555   \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
4556   \setCJKmonofont { FZFangSong-Z02 }
4557   \setCJKfamilyfont { zhsong } { FZShuSong-Z01 }
4558   [ BoldFont = FZXiaoBiaoSong-B05 ]
4559   \setCJKfamilyfont { zhhei } { FZHei-B01 }
4560   \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4561   \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4562   \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4563   \setCJKfamilyfont { zhyou } { FZXiYuan-M01 }
4564   [ BoldFont = FZZhunYuan-M02 ]
4565 }
4566 </founder>

```

14.15.2.4 mac 相关

按 [Issue 351](#) 的讨论, 以 El Capitan 为分界, 分别设置 macold (El Capitan 之前) 和 macnew (El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。

```
4567 <*mac>
4568 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4569 { \ctex_file_input:n { ctex-fontset-macnew.def } }
4570 { \ctex_file_input:n { ctex-fontset-macold.def } }
4571 </mac>
```

macold 的设置参考了 [OS X Mavericks \(10.9\)](#) 预装的主要简体中文字体列表。

macnew 在默认字体设置方面, 引入了多字重的宋体作为罗马字族, 以及引入了苹方黑体作为无衬线字族。由于 Songti SC Light 的字重与 STSong 及 Windows 上的 SimSun 更接近, 故默认字重使用 Songti SC Light, 而不带后缀的正常字重事实上没有使用。黑体、圆体等设置也有类似的情况。

```
4572 <*macold|macnew>
4573 \ctex_fontset_case:nnnn
4574 { \ctex_fontset_error:n { mac } }
4575 <*macold>
4576 { \ctex_fontset_error:n { macold } }
4577 { \ctex_fontset_error:n { macold } }
4578 </macold>
4579 <*macnew>
4580 {
4581   \ctex_zhmap_case:nnn
4582   {
4583     \setCJKmainfont { :3:Songti.ttc }
4584     [
4585       BoldFont      = :1:Songti.ttc,
4586       ItalicFont    = :0:Kaiti.ttc,
4587       BoldItalicFont = :3:Kaiti.ttc,
4588     ]
4589     \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4590     \setCJKmonofont { STFANGSO.ttf }
4591     \setCJKfamilyfont { zhsong } { :3:Songti.ttc } [ BoldFont = :1:Songti.ttc ]
4592     \setCJKfamilyfont { zhhei } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4593     \setCJKfamilyfont { zhkai } { :0:Kaiti.ttc } [ BoldFont = :3:Kaiti.ttc ]
4594     \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
4595     \setCJKfamilyfont { zhli } { :0:Baoli.ttc }
4596     \setCJKfamilyfont { zhyou } { :4:Yuanti.ttc } [ BoldFont = :0:Yuanti.ttc ]
4597     \ctex_punct_set:n { mac }
4598     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4599     \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
4600     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4601     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4602     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4603     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhpfb }
4604     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4605   }
4606   {
4607     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { mac }
4608     \ctex_punct_set:n { mac }
4609     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4610     \ctex_punct_map_family:nn { \CJKsfdefault } { zhpfb }
4611     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhpfb }
4612     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4613   }
4614   { \ctex_fontset_error:n { macnew } }
4615 }
4616 {
4617   \ctex_set_upmap:nnn { upserif } { :3:Songti.ttc } { :1:Songti.ttc }
4618   \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc } { }
4619   \ctex_set_upmap:nnn { upstsl } { :0:Baoli.ttc } { }
```

```

4620 \ctex_set_upmap:nnn { upstht } { :4:Yuanti.ttc } { }
4621 \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
4622 \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf } { }
4623 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4624 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4625 \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
4626 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4627 \ctex_set_upfamily:nnn { zhli } { upschrm } { }
4628 \ctex_set_upfamily:nnn { zhyou } { upschgt } { }
4629 }
4630 </macnew>
4631 {
4632 <*macold>
4633 \setCJKmainfont { STSong }
4634 [ BoldFont = STHeiti, ItalicFont = STKaiti ]
4635 \setCJKsansfont { STXihei } [ BoldFont = STHeiti ]
4636 \setCJKmonofont { STFangsong }
4637 \setCJKfamilyfont { zhsong } { STSong }
4638 \setCJKfamilyfont { zhhei } { STHeiti }
4639 \setCJKfamilyfont { zhfs } { STFangsong }
4640 \setCJKfamilyfont { zhkai } { STKaiti }
4641 </macold>
4642 <*macnew>
4643 \setCJKmainfont { Songti~SC~Light }
4644 [
4645 BoldFont = Songti~SC~Bold,
4646 ItalicFont = Kaiti~SC,
4647 BoldItalicFont = Kaiti~SC~Bold
4648 ]
4649 \setCJKsansfont { PingFang~SC }
4650 \setCJKmonofont { STFangsong }
4651 \setCJKfamilyfont { zhsong } { Songti~SC~Light } [ BoldFont = Songti~SC~Bold ]
4652 \setCJKfamilyfont { zhhei } { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ]
4653 \setCJKfamilyfont { zhpfs } { PingFang~SC }
4654 \setCJKfamilyfont { zhfs } { STFangsong }
4655 \setCJKfamilyfont { zhkai } { Kaiti~SC } [ BoldFont = Kaiti~SC~Bold ]
4656 \setCJKfamilyfont { zhli } { Baoli~SC }
4657 \setCJKfamilyfont { zhyou } { Yuanti~SC~Light } [ BoldFont = Yuanti~SC~Regular ]
4658 </macnew>
4659 }
4660 </macold|macnew>

```

14.15.2.5 ubuntu

```

4661 <*ubuntu>
4662 \ctex_fontset_case:nnnn
4663 { \ctex_fontset_error:n { ubuntu } }
4664 {
4665 \ctex_zhmap_case:nnn
4666 {
4667 \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
4668 [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
4669 \setCJKsansfont { :2:NotoSansCJK-Regular.ttc }
4670 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4671 \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
4672 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
4673 \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
4674 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
4675 \setCJKfamilyfont { zhhei } { :2:NotoSansCJK-Regular.ttc }
4676 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4677 \setCJKfamilyfont { zhkai } { gkai00mp.ttf }
4678 \ctex_punct_set:n { ubuntu }
4679 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4680 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4681 \ctex_punct_map_family:nn { \CJKttdefault } { zhsong }
4682 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }

```

```

4683     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4684     \ctex_punct_map_bfseries:nn
4685     { \CJKrmdefault, \CJKttdefault, zhsong }
4686     { zhsongb }
4687   }
4688   {
4689     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { ubuntu }
4690     \ctex_punct_set:n { ubuntu }
4691     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4692     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4693     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4694   }
4695   { \ctex_fontset_error:n { ubuntu } }
4696 }
4697 {
4698   \ctex_set_upmap_unicode:nnn { upserif }
4699   { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
4700   \ctex_set_upmap_unicode:nnn { upsans }
4701   { :2:NotoSansCJK-Regular.ttc } { :2:NotoSansCJK-Bold.ttc }
4702   \ctex_set_upmap_unicode:nnn { upmono }
4703   { :2:NotoSerifCJK-Regular.ttc } { }
4704   \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf } { }
4705   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4706   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4707   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4708 }
4709 {
4710   \setCJKmainfont { Noto~Serif~CJK~SC } [ ItalicFont = AR~PL~KaitiM~GB ]
4711   \setCJKsansfont { Noto~Sans~CJK~SC }
4712   \setCJKmonofont { Noto~Serif~CJK~SC }
4713   \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
4714   \setCJKfamilyfont { zhhei } { Noto~Sans~CJK~SC }
4715   \setCJKfamilyfont { zhkai } { AR~PL~KaitiM~GB }
4716 }
4717 </ubuntu>

```

14.15.2.6 windows

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

```

4718 <*windows>
4719 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
4720 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4721 {
4722   \file_if_exist:nTF { msyh.ttc }
4723   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4724   { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
4725 }
4726 \ctex_fontset_case:nnn
4727 {
4728   \ctex_zhmap_case:nnn
4729   {
4730     \ctex_punct_set:n { windows }
4731     \setCJKmainfont { simsun.ttc }
4732     [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
4733     \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
4734     [ BoldFont = msykbd.\c__ctex_msyh_suffix_tl ]
4735     \setCJKmonofont { simfang.ttf }
4736     \setCJKfamilyfont { zhsong } { simsun.ttc }
4737     \setCJKfamilyfont { zhhei } { simhei.ttf }
4738     \setCJKfamilyfont { zhfs } { simfang.ttf }
4739     \setCJKfamilyfont { zhkai } { simkai.ttf }
4740     \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
4741     [ BoldFont = msykbd.\c__ctex_msyh_suffix_tl ]
4742     \setCJKfamilyfont { zhli } { simli.ttf }
4743     \setCJKfamilyfont { zhyou } { simyou.ttf }
4744     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }

```

```

4745 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4746 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4747 \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4748 \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
4749 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4750 }
4751 {
4752 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { windows }
4753 \ctex_punct_set:n { windows }
4754 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4755 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4756 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4757 }
4758 { }
4759 }
4760 {
4761 \ctex_set_upfonts:nnnnnn
4762 { simsun.ttc }
4763 { simhei.ttf }
4764 { simkai.ttf }
4765 { msyh.\c__ctex_msyh_suffix_tl }
4766 { msyhbd.\c__ctex_msyh_suffix_tl }
4767 { simfang.ttf }
4768 \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4769 \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4770 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4771 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4772 \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4773 \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4774 \ctex_set_upfamily:nnn { zhyou } { upscht } {}
4775 \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4776 \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4777 }
4778 {
4779 \setCJKmainfont { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
4780 \setCJKsansfont { MicrosoftYaHei } [ BoldFont = *Bold ]
4781 \setCJKmonofont { FangSong }
4782 \setCJKfamilyfont { zhsong } { SimSun }
4783 \setCJKfamilyfont { zhhei } { SimHei }
4784 \setCJKfamilyfont { zhfs } { FangSong }
4785 \setCJKfamilyfont { zhkai } { KaiTi }
4786 \setCJKfamilyfont { zhyahei } { MicrosoftYaHei } [ BoldFont = *Bold ]
4787 \setCJKfamilyfont { zhli } { LiSu }
4788 \setCJKfamilyfont { zhyou } { YouYuan }
4789 }
4790 </windows>

```

14.15.3 中文字体命令

使用 upL^AT_EX 编译时, macnew 字库中由于传统黑体(黑体-简)无法使用, 我们用苹方来代替。同时 \yahei、\pingfang 命令被设置为与 \heiti 相同。

```

\songti
\heiti
\fangsong
\kaishu
\lishu
\youyuan
\yahei
\pingfang
4791 <!*mac>
4792 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }
4793 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
4794 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4795 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4796 <*windows|founder|macnew>
4797 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4798 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4799 </windows|founder|macnew>
<windows> 4800 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4801 <*macnew>
4802 \bool_lazy_or:nnTF
4803 { \sys_if_engine_pdftex_p: }

```



```

4804 { \sys_if_engine_uptex_p: }
4805 {
4806   \cs_new_eq:NN \yahei \heiti
4807   \cs_new_eq:NN \pingfang \heiti
4808 }
4809 {
4810   \NewDocumentCommand \yahei { } { \CJKfamily { zhp } }
4811   \NewDocumentCommand \pingfang { } { \CJKfamily { zhp } }
4812 }
4813 </macnew>
4814 </!mac>

4815 </fontset>

```

14.15.4 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4816 <*zhmap>
4817 \begingroup\catcode61\catcode48\catcode32=10\relax%
4818 \catcode 35=6 % #
4819 \catcode 45=12 % -
4820 \catcode123=1 % {
4821 \catcode125=2 % }
4822 \toks0{\endlinechar=\the\endlinechar\relax}%
4823 \toks2{\endlinechar=-1}%
4824 \def\x#1 #2 {%
4825   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4826   \toks2\expandafter{\the\toks2 \catcode#1=#2 }}%
4827 \x 13 5 % carriage return
4828 \x 32 10 % space
4829 \x 35 6 % #
4830 \x 40 12 % (
4831 \x 41 12 % )
4832 \x 45 12 % -
4833 \x 46 12 % .
4834 \x 47 12 % /
4835 \x 58 12 % :
4836 \x 60 12 % <
4837 \x 61 12 % =
4838 \x 64 11 % @
4839 \x 91 12 % [
4840 \x 93 12 % ]
4841 \x 123 1 % {
4842 \x 125 2 % }
4843 \edef\x#1{\endgroup%
4844   \edef\noexpand#1{%
4845     \the\toks0 %
4846     \let\noexpand\noexpand\noexpand#1%
4847     \noexpand\noexpand\noexpand\undefined%
4848     \noexpand\noexpand\noexpand\endinput}%
4849     \the\toks2}%
4850 \expandafter\x\csname ctex@zhmap@endinput\endcsname

4851 \begingroup\expandafter\endgroup
4852 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname
4853   \expandafter\ifx\csname ifctexpdf\endcsname\relax
4854     \expandafter\ifx\csname pdfoutput\endcsname\relax
4855       iffalse\else\ifnum\pdfoutput < 1 iffalse\else iftrue\fi\fi
4856     \else ifctexpdf\fi
4857 \endcsname

\ProvidesFile 提供非 LATEX 格式下的 \ProvidesFile。

4858 \begingroup
4859 \expandafter\ifx\csname ProvidesFile\endcsname\relax

```



```

4860 \long\def\x#1\ProvidesFile#2[#3]{%
4861   #1%
4862   \immediate\write-1{File: #2 #3}%
4863   \expandafter\xdef\csname ver@#2\endcsname{#3}}
4864 \expandafter\x%
4865 \fi
4866 \endgroup

```

文件标识信息。

```

<adobe> 4867 \ProvidesFile{ctex-zhmap-adobe.tex}%
<adobe> 4868 [2021/03/14 v2.5.6 Adobe font map loader for DVIPDFMx (CTEX)]
<fandol> 4869 \ProvidesFile{ctex-zhmap-fandol.tex}%
<fandol> 4870 [2021/03/14 v2.5.6 Fandol font map loader for DVIPDFMx (CTEX)]
<founder> 4871 \ProvidesFile{ctex-zhmap-founder.tex}%
<founder> 4872 [2021/03/14 v2.5.6 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
<mac> 4873 \ProvidesFile{ctex-zhmap-mac.tex}%
<mac> 4874 [2021/03/14 v2.5.6 Mac font map loader for DVIPDFMx (CTEX)]
<ubuntu> 4875 \ProvidesFile{ctex-zhmap-ubuntu.tex}%
<ubuntu> 4876 [2021/03/14 v2.5.6 Ubuntu font map loader for DVIPDFMx (CTEX)]
<windows> 4877 \ProvidesFile{ctex-zhmap-windows.tex}%
<windows> 4878 [2021/03/14 v2.5.6 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]

```

14.15.4.1 ctex-zhmap-adobe.tex

```

4879 <*adobe>
4880 \ifzhmappdf
4881 %% pdfTeX does not support OTF fonts
4882 \else
4883 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4884 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4885 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4886 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4887 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4888 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4889 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4890 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4891 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4892 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4893 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4894 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4895 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4896 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4897 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4898 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4899 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4900 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4901 \fi
4902 </adobe>

```

14.15.4.2 ctex-zhmap-fandol.tex

```

4903 <*fandol>
4904 \ifzhmappdf
4905 %% pdfTeX does not support OTF fonts
4906 \else
4907 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4908 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4909 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4910 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4911 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4912 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4913 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4914 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4915 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4916 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4917 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}

```

```

4918 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4919 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4920 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4921 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4922 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4923 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4924 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4925 \fi
4926 </fandol>

```

14.15.4.3 ctex-zhmap-founder.tex

```

4927 <*founder>
4928 \ifzhmappdf
4929 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4930 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4931 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4932 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4933 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4934 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4935 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4936 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4937 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4938 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4939 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4940 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4941 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4942 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4943 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4944 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4945 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4946 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4947 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4948 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4949 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4950 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4951 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4952 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4953 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4954 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4955 \else
4956 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4957 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4958 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4959 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4960 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4961 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4962 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4963 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4964 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4965 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4966 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4967 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4968 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4969 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4970 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4971 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4972 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}
4973 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4974 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4975 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4976 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4977 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4978 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4979 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4980 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4981 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4982 \fi

```

4983 </founder>

14.15.4.4 ctex-zhmap-mac.tex

```

4984 <*mac>
4985 \ifzhmappdf
4986 %% pdfTeX does not support OTF fonts
4987 \else
4988 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4989 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4990 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc}
4991 \special{pdf:mapline gbkhei@UGBK@ unicode :2:PingFang.ttc}
4992 \special{pdf:mapline gbkfs@UGBK@ unicode STFANGSO.ttf}
4993 \special{pdf:mapline gbkli@UGBK@ UniGB-UTF16-H :0:Baoli.ttc}
4994 \special{pdf:mapline gbkyou@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc}
4995 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4996 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4997 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc}
4998 \special{pdf:mapline unihei@Unicode@ unicode :2:PingFang.ttc}
4999 \special{pdf:mapline unifs@Unicode@ unicode STFANGSO.ttf}
5000 \special{pdf:mapline unili@Unicode@ UniGB-UTF16-H :0:Baoli.ttc}
5001 \special{pdf:mapline uniyou@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc}
5002 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H :3:Songti.ttc -s .167}
5003 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
5004 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:PingFang.ttc -s .167}
5005 \special{pdf:mapline gbkfssl@UGBK@ unicode STFANGSO.ttf -s .167}
5006 \special{pdf:mapline gbklisl@UGBK@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
5007 \special{pdf:mapline gbkyousl@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
5008 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc -s .167}
5009 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
5010 \special{pdf:mapline uniheisl@Unicode@ unicode :2:PingFang.ttc -s .167}
5011 \special{pdf:mapline unifssl@Unicode@ unicode STFANGSO.ttf -s .167}
5012 \special{pdf:mapline unilisl@Unicode@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
5013 \special{pdf:mapline uniyousl@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
5014 \fi
5015 </mac>

```

14.15.4.5 ctex-zhmap-ubuntu.tex

```

5016 <*ubuntu>
5017 \ifzhmappdf
5018 %% pdfTeX does not support OTF fonts
5019 \else
5020 \special{pdf:mapline gbk@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5021 \special{pdf:mapline gbksong@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5022 \special{pdf:mapline gbkkai@UGBK@ unicode gkai00mp.ttf}
5023 \special{pdf:mapline gbkhei@UGBK@ unicode :2:NotoSansCJK-Regular.ttc}
5024 \special{pdf:mapline gbkfs@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
5025 \special{pdf:mapline cyberb@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5026 \special{pdf:mapline unisong@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5027 \special{pdf:mapline unikai@Unicode@ unicode gkai00mp.ttf}
5028 \special{pdf:mapline unihei@Unicode@ unicode :2:NotoSansCJK-Regular.ttc}
5029 \special{pdf:mapline unifs@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
5030 \special{pdf:mapline gbksongsl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5031 \special{pdf:mapline gbkkaisl@UGBK@ unicode gkai00mp.ttf -s .167}
5032 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
5033 \special{pdf:mapline gbkfssl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5034 \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5035 \special{pdf:mapline unikaisl@Unicode@ unicode gkai00mp.ttf -s .167}
5036 \special{pdf:mapline uniheisl@Unicode@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
5037 \special{pdf:mapline unifssl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
5038 \fi
5039 </ubuntu>

```

14.15.4.6 ctex-zhmap-windows.tex

```

5040 <*windows>
5041 \ifzhmappdf

```

```

5042 \pdfmapline{=gbk@UGBK@ <simsun.ttc}
5043 \pdfmapline{=gbksong@UGBK@ <simsun.ttc}
5044 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
5045 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
5046 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
5047 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
5048 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
5049 \pdfmapline{=cyberb@Unicode@ <simsun.ttc}
5050 \pdfmapline{=unisong@Unicode@ <simsun.ttc}
5051 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
5052 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
5053 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
5054 \pdfmapline{=unili@Unicode@ <simli.ttf}
5055 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
5056 \pdfmapline{=gbksongsl@UGBK@ <simsun.ttc}
5057 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
5058 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
5059 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
5060 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
5061 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
5062 \pdfmapline{=unisongsl@Unicode@ <simsun.ttc}
5063 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
5064 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
5065 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
5066 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
5067 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
5068 \else
5069 \special{pdf:mapline gbk@UGBK@ unicode :0:simsun.ttc -v 50}
5070 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsun.ttc -v 50}
5071 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
5072 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
5073 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
5074 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
5075 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
5076 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsun.ttc -v 50}
5077 \special{pdf:mapline unisong@Unicode@ unicode :0:simsun.ttc -v 50}
5078 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
5079 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
5080 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
5081 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
5082 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
5083 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsun.ttc -v 50 -s .167}
5084 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -v 70 -s .167}
5085 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -v 150 -s .167}
5086 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -v 50 -s .167}
5087 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -v 150 -s .167}
5088 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -v 60 -s .167}
5089 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsun.ttc -v 50 -s .167}
5090 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -v 70 -s .167}
5091 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -v 150 -s .167}
5092 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -v 50 -s .167}
5093 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -v 150 -s .167}
5094 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -v 60 -s .167}
5095 \fi
5096 </windows>
5097 \ctex@zhmap@endinput
5098 </zhmap>

```

14.15.5 制作 spa 文件

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

```

5099 <*spa>
5100 <*macro>
5101 \input expl3-generic %

```

```

5102 \ExplSyntaxOn
5103 \sys_if_engine_xetex:F
5104 {
5105   \msg_new:nnn { ctex } { xetex }
5106   { XeTeX~is~required~to~compile~this~document! }
5107   \msg_fatal:nn { ctex } { xetex }
5108 }

```

CJKpunct 定义的标点符号是:

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

注意顺序不能改变。

```

5109 \seq_const_from_clist:Nn \c__ctex_punct_seq
5110 {
5111   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
5112   "3008 , "300A , "3016 , "3010 ,
5113   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
5114   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
5115   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
5116 }

```

\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。

```

5117 \cs_new_protected:Npn \ctex_make_spa:nn #1#2
5118 {
5119   \iow_open:Nn \g__ctex_spa_iow {#1}
5120   \clist_map_inline:nn {#2}
5121   { \__ctex_write_family:nn ##1 }
5122   \iow_close:N \g__ctex_spa_iow
5123 }
5124 \iow_new:N \g__ctex_spa_iow
5125 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn

5126 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
5127 {
5128   \group_begin:
5129   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
5130   \l__ctex_punct_font
5131   \clist_clear:N \l__ctex_punct_bounds_clist
5132   \seq_map_inline:Nn \c__ctex_punct_seq
5133   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \tex_XeTeXcharglyph:D ##1 } }
5134   \iow_now:Nx \g__ctex_spa_iow
5135   {
5136     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

5137     { \l__ctex_punct_bounds_clist , , , }
5138   }
5139   \group_end:
5140 }
5141 \cs_new_protected:Npn \__ctex_save_bounds:n #1
5142 {
5143   \clist_put_right:Nx \l__ctex_punct_bounds_clist
5144   {
5145     \__ctex_calc_bounds:nn { 1 } {#1} ,
5146     \__ctex_calc_bounds:nn { 3 } {#1}
5147   }
5148 }
5149 \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。

```

5150 \cs_new:Npn \__ctex_calc_bounds:nn #1#2
5151 {
5152   \fp_eval:n

```

```

5153     {
5154         round
5155         (
5156             \dim_to_decimal_in_unit:nn
5157             { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
5158             { 1 em }
5159         )
5160     }
5161 }
5162 \ExplSyntaxOff
5163 </macro>

```

下面是 CTeX 定义的一些字体。

```

5164 <*make>
5165 \input ctex-spa-macro %
5166 \MAKESPA {ctexpunct.spa}
5167 {
5168     {adobezhsong}      {AdobeSongStd-Light} ,
5169     {adobezhhei}       {AdobeHeitiStd-Regular} ,
5170     {adobezhkai}       {AdobeKaitiStd-Regular} ,
5171     {adobezhfs}        {AdobeFangsongStd-Regular} ,
5172 %
5173     {fandolzhsong}     {FandolSong} ,
5174     {fandolzhsongb}    {FandolSong-Bold} ,
5175     {fandolzhhei}      {FandolHei} ,
5176     {fandolzhheib}     {FandolHei-Bold} ,
5177     {fandolzhkai}      {FandolKai} ,
5178     {fandolzhfs}       {FandolFang} ,
5179 %
5180     {founderzhsong}    {FZShuSong-Z01} ,
5181     {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
5182     {founderzhhei}     {FZHei-B01} ,
5183     {founderzhheil}    {FZXiHeiI-Z08} ,
5184     {founderzhkai}     {FZKai-Z03} ,
5185     {founderzhfs}      {FZFangSong-Z02} ,
5186     {founderzhli}      {FZLiShu-S01} ,
5187     {founderzhyou}     {FZXiYuan-M01} ,
5188     {founderzhyoub}    {FZZhunYuan-M02} ,
5189 %
5190     {maczhsong}        {Songti SC Light} ,
5191     {maczhsongb}       {Songti SC Bold} ,
5192     {maczhhei}         {Heiti SC Medium} ,
5193     {maczhheil}        {Heiti SC Light} ,
5194     {maczhkai}         {Kaiti SC} ,
5195     {maczhkaib}        {Kaiti SC Bold} ,
5196     {maczhfs}          {STFangsong} ,
5197     {maczhli}          {Baoli SC} ,
5198     {maczhyou}         {Yuanti SC Light} ,
5199     {maczhyoub}        {Yuanti SC Regular} ,
5200     {maczhpf}          {PingFang SC} ,
5201     {maczhpfb}         {PingFang SC Semibold} ,
5202 %
5203     {ubuntuzhsong}     {Noto Serif CJK SC} ,
5204     {ubuntuzhsongb}    {Noto Serif CJK SC Bold} ,
5205     {ubuntuzhhei}      {Noto Sans CJK SC} ,
5206     {ubuntuzhheib}     {Noto Sans CJK SC Bold} ,
5207     {ubuntuzhkai}      {AR PL KaitiM GB} ,
5208 %
5209     {windowszhsong}    {SimSun} ,
5210     {windowszhhei}     {SimHei} ,
5211     {windowszhkai}     {KaiTi} ,
5212     {windowszhfs}      {FangSong} ,
5213     {windowszhli}      {LiSu} ,
5214     {windowszhyou}     {YouYuan} ,
5215     {windowszhyahei}   {Microsoft YaHei} ,
5216     {windowszhyaheib} {Microsoft YaHei Bold}
5217 }

```



```

5218 \primitive\end
5219 </make>
5220 </spa>

```

14.16 translator 宏包的中文字典

```

5221 <*dict>

```

包括 ChineseGBK 和 ChineseUTF8 两种形式, 目前只翻译 beamer 宏包需要的定理环境名称。

```

5222 <*theorem>
5223 \providetranslation{Comments}{评论}
5224 \providetranslation{comments}{评论}
5225 \providetranslation{Comment}{评论}
5226 \providetranslation{comment}{评论}
5227 \providetranslation{Corollaries}{推论}
5228 \providetranslation{corollaries}{推论}
5229 \providetranslation{Corollary}{推论}
5230 \providetranslation{corollary}{推论}
5231 \providetranslation{Definitions}{定义}
5232 \providetranslation{definitions}{定义}
5233 \providetranslation{Definition}{定义}
5234 \providetranslation{definition}{定义}
5235 \providetranslation{Examples}{例}
5236 \providetranslation{examples}{例}
5237 \providetranslation{Example}{例}
5238 \providetranslation{example}{例}
5239 \providetranslation{Exercises}{练习}
5240 \providetranslation{exercises}{练习}
5241 \providetranslation{Exercise}{练习}
5242 \providetranslation{exercise}{练习}
5243 \providetranslation{Facts}{事实}
5244 \providetranslation{facts}{事实}
5245 \providetranslation{Fact}{事实}
5246 \providetranslation{fact}{事实}
5247 \providetranslation{Key Lemmas}{关键引理}
5248 \providetranslation{key lemmas}{关键引理}
5249 \providetranslation{Key Lemma}{关键引理}
5250 \providetranslation{key lemma}{关键引理}
5251 \providetranslation{Key Observations}{关键观察}
5252 \providetranslation{key observations}{关键观察}
5253 \providetranslation{Key Observation}{关键观察}
5254 \providetranslation{key observation}{关键观察}
5255 \providetranslation{Lemmas}{引理}
5256 \providetranslation{lemmas}{引理}
5257 \providetranslation{Lemma}{引理}
5258 \providetranslation{lemma}{引理}
5259 \providetranslation{Main Theorems}{主要定理}
5260 \providetranslation{main theorems}{主要定理}
5261 \providetranslation{Main Theorem}{主要定理}
5262 \providetranslation{main theorem}{主要定理}
5263 \providetranslation{Observations}{观察}
5264 \providetranslation{observations}{观察}
5265 \providetranslation{Observation}{观察}
5266 \providetranslation{observation}{观察}
5267 \providetranslation{Problems}{问题}
5268 \providetranslation{problems}{问题}
5269 \providetranslation{Problem}{问题}
5270 \providetranslation{problem}{问题}
5271 \providetranslation{Proofs}{证明}
5272 \providetranslation{proofs}{证明}
5273 \providetranslation{Proof}{证明}
5274 \providetranslation{proof}{证明}
5275 \providetranslation{Proof Sketch}{证明提要}
5276 \providetranslation{Proof sketch}{证明提要}

```

```

5277 \providetranslation{proof sketch}{证明提要}
5278 \providetranslation{Proof Sketches}{证明提要}
5279 \providetranslation{Proof sketches}{证明提要}
5280 \providetranslation{proof sketches}{证明提要}
5281 \providetranslation{Sketch of Proof}{证明提要}
5282 \providetranslation{Sketch of Proofs}{证明提要}
5283 \providetranslation{Sketch of proof}{证明提要}
5284 \providetranslation{Sketch of proofs}{证明提要}
5285 \providetranslation{sketch of proof}{证明提要}
5286 \providetranslation{sketch of proofs}{证明提要}
5287 \providetranslation{Propositions}{命题}
5288 \providetranslation{propositions}{命题}
5289 \providetranslation{Proposition}{命题}
5290 \providetranslation{proposition}{命题}
5291 \providetranslation{Remarks}{注}
5292 \providetranslation{remarks}{注}
5293 \providetranslation{Remark}{注}
5294 \providetranslation{remark}{注}
5295 \providetranslation{Solutions}{解}
5296 \providetranslation{solutions}{解}
5297 \providetranslation{Solution}{解}
5298 \providetranslation{solution}{解}
5299 \providetranslation{Theorems}{定理}
5300 \providetranslation{theorems}{定理}
5301 \providetranslation{Theorem}{定理}
5302 \providetranslation{theorem}{定理}
5303 </theorem>
5304 </dict>

```

14.17 ctexcap 宏包

5305 <*ctexcap>

ctexcap 是过时宏包。

```

5306 \clist_new:N \l__ctex_ctexcap_options_clist
5307 \clist_set:Nx \l__ctex_ctexcap_options_clist
5308 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
5309 \msg_new:nnn { ctexcap } { deprecated }
5310 {
5311   Package~`ctexcap'~is~deprecated.\\
5312   Please~use~package~`ctex'~with~option~`#1'~instead: \\\\
5313   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5314 }
5315 \msg_warning:nnx { ctexcap } { deprecated }
5316 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5317 \PassOptionsToPackage { heading = true } { ctexcap }
5318 \RequirePackageWithOptions { ctex }
5319 </ctexcap>

```

14.18 ctexhook 宏包

5320 <*ctexhook>

\ctex_at_begin_package:nn 如果宏包已经被载入,则钩子无效,给出警告。

```

5321 \cs_new_protected:Npn \ctex_at_begin_package:nn #1
5322 {
5323   \@ifpackageloaded {#1}
5324   { \__ctex_package_loaded_warning:nn {#1} }
5325   { \ctex_gadd_package_hook:nnn { before } {#1} }
5326 }
5327 \cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
5328 { \msg_warning:nnx { ctex } { invalid-hook } {#1} }
5329 \msg_new:nnn { ctex } { invalid-hook }

```



```

5330 {
5331   Package~`#1'~is~loaded. \\
5332   \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
5333 }

```

\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。

```

5334 \cs_new_protected:Npn \ctex_at_end_package:nn #1
5335 {
5336   \@ifpackageloaded {#1}
5337   { \use:n }
5338   { \ctex_gadd_package_hook:nnn { after } {#1} }
5339 }

```

L^AT_EX 2020/10/01 开始提供常用钩子管理机制。在新机制下,我们只需要做简单的包装。

```

5340 \@ifl@t@r \fmtversion { 2020/10/01 }
5341 {
5342   \cs_new_protected:Npx \ctex_gadd_ltxhook:nn #1
5343   { \hook_gput_code:nnn {#1} { \c_novalue_tl } }
5344   \cs_new_protected:Npn \ctex_at_end_preamble:n
5345   { \ctex_gadd_ltxhook:nn { begindocument/before } }
5346   \cs_new_protected:Npn \ctex_after_end_preamble:n
5347   { \ctex_gadd_ltxhook:nn { begindocument/end } }
5348   \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
5349   { \ctex_gadd_ltxhook:nn { package/#1/#2 } }
5350   \file_input_stop:
5351 }
5352 { }

```

对于 L^AT_EX 2020/10/01 之前的版本,需要自行补丁。

\ctex_at_end_preamble:n 实现 etoolbox 宏包的 \AtEndPreamble 和 \AfterEndPreamble。
 \ctex_after_end_preamble:n

```

5353 \cs_new_protected:Npn \ctex_at_end_preamble:n
5354 { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
5355 \cs_new_protected:Npn \ctex_after_end_preamble:n
5356 { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
5357 \cs_new_protected:Npn \CTEX@document@left@hook
5358 { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5359 \cs_new_protected:Npn \CTEX@document@right@hook
5360 { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5361 \cs_set_nopar:Npx \document
5362 {
5363   \CTEX@document@left@hook
5364   \exp_not:o { \document }
5365   \CTEX@document@right@hook
5366 }
5367 \tl_new:N \g__ctex_end_preamble_hook_tl
5368 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

\ctex_gadd_package_hook:nnn 给钩子附加内容。

```

\ctex_gadd_hook:Nn
\ctex_gadd_hook:cn
5369 \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
5370 { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }
5371 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1
5372 {
5373   \tl_if_exist:NF #1 { \tl_new:N #1 }
5374   \tl_gput_right:Nn #1
5375 }
5376 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

\ctex_use_package_hook:nn 宏包钩子,只执行一次,用后清除。

```

5377 \cs_new_protected:Npn \ctex_use_package_hook:nn #1#2

```

```

5378 {
5379   \group_begin: \exp_args:Nnc \group_end:
5380   \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
5381 }
5382 \cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
5383 { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }

```

\@resetOptions
\CTEX@resetOptions@hook

\@pushfilename 内部的 \@currname 和 \@currentx 保存的是前一个宏包的状态,不能使用。
需要对其后的 \@resetOptions 做补丁来实现 \ctex_at_begin_package:nn 的功能。

```

5384 \tl_put_right:Nn \@resetOptions { \CTEX@resetOptions@hook }
5385 \cs_new_protected:Npn \CTEX@resetOptions@hook
5386 {
5387   \cs_if_eq:NNT \@currentx \pkgextension
5388   { \ctex_use_package_hook:nn { before } { \@currname } }
5389 }

```

\@popfilename
\CTEX@popfilename@hook

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

```

5390 \tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
5391 \cs_new_protected:Npn \CTEX@popfilename@hook
5392 {
5393   \cs_if_eq:NNT \@currentx \pkgextension
5394   { \ctex_use_package_hook:nn { after } { \@currname } }
5395 }

```

5396 </ctexhook>

14.19 ctexpatch 宏包

5397 <*ctexpatch>

\ctex_patch_cmd_once:NnnnTF

只进行第一次匹配进行替换。参数 #2 是宏重建时的 \catcode 设置。

```

5398 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5399 {
5400   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5401   { once } {#2} { \use_i:nn } { \use_ii:nn }
5402 }

```

\ctex_patch_cmd_all:NnnnTF

替换所有匹配到的文本。

```

5403 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5404 {
5405   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5406   { all } {#2} { \use_i:nn } { \use_ii:nn }
5407 }

```

\ctex_patch_cmd:Nnn

快捷方式,在补丁的时候关闭 L^AT_EX3 语法和设置 @ 为字母类,补丁失败时给出警告。

```

5408 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5409 {
5410   \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5411   { once }
5412   {
5413     \ExplSyntaxOff
5414     \char_set_catcode_letter:n { 64 }
5415   }
5416   { }
5417   { \ctex_patch_failure:N #1 }
5418 }
5419 \cs_new_protected:Npn \ctex_patch_failure:N #1
5420 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5421 \msg_new:nnn { ctex } { patch-failure }
5422 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

`\ctex_preto_cmd:NnnTF` 在宏的原本定义前面增加钩子。

```
5423 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5424 {
5425   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5426   { left } {#2} { \use_i:nn } { \use_ii:nn }
5427 }
```

`\ctex_appto_cmd:NnnTF` 在宏的原本定义后面追加钩子。

```
5428 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5429 {
5430   \ctex_patch_boot:NNnnTF \__ctex_hookto_cmd:Nnnnw #1
5431   { right } {#2} { \use_i:nn } { \use_ii:nn }
5432 }
```

`\ctex_patch_boot:NNnnTF` 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。

```
5433 \cs_new_protected:Npn \ctex_patch_boot:NNnnTF #1#2#3#4#5#6
5434 {
5435   \cs_set_protected:Npx \__ctex_patch_true:w { \exp_not:n {#5} }
5436   \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
5437   \group_begin:
5438     \char_set_catcode_other:n { 35 }
5439     \ctex_parse_name:NN #1 #2 {#3} {#4}
5440   }
5441   \cs_new_eq:NN \__ctex_patch_true:w \use_i:nn
5442   \cs_new_eq:NN \__ctex_patch_false:w \use_ii:nn
```

`\ctex_parse_name:NN` 用 `\DeclareRobustCommand` 定义的宏或者由 `\newcommand` 或 `\newrobustcmd` 定义的带一个可选参数的宏第一次展开的结果都不是其实际定义,实际定义被保存在另外的宏中。由这些命令定义的宏的第一次展开结果可以有下面的形式(细节可查阅 `xpatch` 的文档):

```
1 \protect_\xaa_\_          % \DeclareRobustCommand\xaa[1]{...}
2 \protect_\xab_\_          % \DeclareRobustCommand\xab[1] [] {...}
3 \@protected@testopt_\xac_\_ \xac_\_ { } % \newcommand\xac[1] [] {...}
4 \@testopt_\xad_\_          % \newrobustcmd\xad[1] [] {...}
5 \x@protect_\1\protect_\1_\_ % \DeclareRobustCommand\1[1]{...}
6 \x@protect_\2\protect_\2_\_ % \DeclareRobustCommand\2[1] [] {...}
7 \@protected@testopt_\3\3_\_ { } % \newcommand\3[1] [] {...}
8 \@testopt_\4_\_          % \newrobustcmd\4[1] [] {...}
```

`ctexpatch` 的主要原理是先对宏的 `\meaning` 作字符串替换,然后再用 `\scantokens` 来重建它。我们希望对宏的实际定义打补丁,为此需要先得到对应的名字。`letltxmacro`、`show2e` 和 `xpatch` 宏包中都有类似的工作。

```
5443 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5444 { \ctex_parse_name:NNx #1#2 { \cs_to_str:N #2 } }
5445 \group_begin:
5446 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5447 {
5448   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5449   {
5450     \bool_lazy_or:nnTF
5451     { \cs_if_exist_p:c { ##3 ~ } }
5452     { \cs_if_exist_p:c { #1##3 } }
5453     {
5454       \group_begin:
5455       \use:x
5456       {
5457         \group_end:
5458         \__ctex_parse_name:nNNNnN
5459         { \cs_replacement_spec:N ##2 }
5460         \exp_not:N ##2
```

```

5461         \exp_not:c { ##3 ~ }
5462         \exp_not:c { #1##3 }
5463     } { ##3 } ##1
5464 }
5465 { ##1##2 }
5466 }
5467 \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5468 {
5469     \exp_args:Nc ##6
5470     {
5471         \str_case:nnTF {##1}
5472         {
5473             { \protect ##3 } { }
5474             { \x@protect ##2 \protect ##3 } { }
5475         }
5476         {
5477             \str_if_eq:eeTF
5478             { \exp_not:n { #1@protected@ ##3 #1##3 } }
5479             {
5480                 \exp_last_unbraced:Ne \__ctex_parse_name:w
5481                 { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
5482             }
5483             { #1##5 ~ } { ##5 ~ }
5484         }
5485         {
5486             \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5487             {
5488                 { #1@protected@ ##2 ##4 } { }
5489                 { #1@ ##4 } { }
5490             }
5491             { #1##5 } { ##5 }
5492         }
5493     }
5494 }
5495 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5496 }
5497 \use:x
5498 {
5499     \__ctex_tmp:w
5500     { \c_backslash_str }
5501     { \c_left_brace_str }
5502     { \tl_to_str:n { testopt } }
5503 }
5504 \group_end:
5505 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 `\meaning` 中的前缀、参数文本和替换文本。

```

5506 \str_new:N \l__ctex_prefix_str
5507 \str_new:N \l__ctex_parameter_str
5508 \str_new:N \l__ctex_replacement_str

```

```

\ctex_get_macro_meaning:NTF
\__ctex_get_macro_meaning:w

```

解构待补丁宏的 `\meaning`。若命令不是宏,则走向 `false` 分支。

```

5509 \group_begin:
5510 \cs_set_protected:Npn \__ctex_tmp:w #1
5511 {
5512     \prg_new_protected_conditional:Npnn
5513     \ctex_get_macro_meaning:N ##1 { TF }
5514     {
5515         \exp_after:wN \__ctex_get_macro_meaning:w
5516         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5517     }
5518     \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5519     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5520     {
5521         \tl_if_empty:nTF { ##4 }

```

```

5522         { \prg_return_false: }
5523         {
5524             \str_set:Nn \l__ctex_prefix_str      { ##1 }
5525             \str_set:Nn \l__ctex_parameter_str   { ##2 }
5526             \str_set:Nn \l__ctex_replacement_str { ##3 }
5527             \prg_return_true:
5528         }
5529     }
5530 }
5531 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5532 \group_end:

```

\ctex_if_rescanable:NnTF 检查宏是否可以重建。

```

5533 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5534 {
5535     \ctex_get_macro_meaning:NnTF #1
5536     {
5537         \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5538         \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5539     }
5540     {#4}
5541 }
5542 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

```

__ctex_patch_rebuild:Nn 使用 \tl_rescan:nn 来重新记号化 \meaning 字符串。

```

5543 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5544 {
5545     \__ctex_patch_rescan:NNn \l__ctex_prefix_tl      \l__ctex_prefix_str      {#2}
5546     \__ctex_patch_rescan:NNn \l__ctex_parameter_tl   \l__ctex_parameter_str   {#2}
5547     \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5548     \use:x
5549     {
5550         \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5551         \exp_not:o { \l__ctex_parameter_tl }
5552         { \exp_not:o { \l__ctex_replacement_tl } }
5553     }
5554 }
5555 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5556 {
5557     \str_if_empty:NnTF #2
5558     { \tl_clear:N #1 }
5559     { \tl_set_rescan:Nno #1 {#3} {#2} }
5560 }
5561 \tl_new:N \l__ctex_prefix_tl
5562 \tl_new:N \l__ctex_parameter_tl
5563 \tl_new:N \l__ctex_replacement_tl

```

__ctex_patch_cmd:Nnnnw 对宏的替换文本进行字符串替换,然后重建。

```

5564 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5565 {
5566     \group_end:
5567     \ctex_if_rescanable:NnTF #1 {#3}
5568     {
5569         \use:x
5570         {
5571             \__ctex_patch_replace:nnnTF {#2}
5572             { \tl_to_str:n {#4} }
5573             { \tl_to_str:n {#5} }
5574         }
5575         {
5576             \__ctex_patch_rebuild:Nn #1 {#3}
5577             \__ctex_patch_true:w
5578         }
5579         { \__ctex_patch_false:w }

```

```

5580     }
5581     { \__ctex_patch_false:w }
5582 }

```

__ctex_patch_replace:nnnTF 替换前先检查原文本是否存在。

```

5583 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5584 {
5585     \tl_if_in:NnTF \l__ctex_replacement_str {#2}
5586     { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5587 }

```

__ctex_hookto_cmd:Nnnnw 在宏的前/后附加钩子。

```

5588 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5589 {
5590     \group_end:
5591     \ctex_get_macro_meaning:NTF #1
5592     {
5593         \str_if_empty:NTF \l__ctex_parameter_str
5594         { \__ctex_hookto_cmd_parameterless:Nnnnw }
5595         { \__ctex_hookto_cmd_parameter:Nnnnw }
5596         #1 {#2} {#3} {#4}
5597     }
5598     { \__ctex_patch_false:w }
5599 }

```

__ctex_hookto_cmd_parameterless:Nnnnw 如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

5600 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5601 {
5602     \str_if_empty:NF \l__ctex_prefix_str
5603     { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5604     \tex_edef:D #1
5605     {
5606         \use:c { __ctex_ #2 _hook_aux:nn }
5607         { \exp_not:o {#1} }
5608         { \exp_not:n {#4} }
5609     }
5610     \__ctex_patch_true:w
5611 }
5612 \cs_generate_variant:Nn \tl_rescan:nn { no }
5613 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5614 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

__ctex_hookto_cmd_parameter:Nnnnw 如果宏有参数,需要在字符串中进行附加,然后再重建。

```

5615 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5616 {
5617     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5618     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5619     {
5620         \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5621         \__ctex_patch_rebuild:Nn #1 {#3}
5622         \__ctex_patch_true:w
5623     }
5624     { \__ctex_patch_false:w }
5625 }
5626 </ctexpatch>

```

版本历史

v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。	42
captiondelimiter 是过时选项。	120
fancyhdr 成为过时选项,原选项功能总是打开。	44
fntef 成为过时选项,原选项功能总是打开。	44
hyperref 成为过时选项,原选项功能总是打开。	44
indent, noindent 是过时选项。	42
nofonts, adobefonts, winfonts 是过时选项。	43
nopunct 是过时选项。	43
nospace 是过时选项。	44
nozhmap 是过时选项。	43
punct 选项可以设置标点格式。	43
ctex 宏包新增 heading 选项。	44
\CTEXindent, \CTEXnoindent 是过时命令。	86
\CTEXsetup, \CTEXoptions 是过时命令。	15, 83
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; \CTEXfilltwosides 是过时环境。	79
标题设置新增 pagestyle 选项。	90
调整 \footnotesep 的大小,以适合行距的变化。	119
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。	45
将标题汉化功能加入 ctex.sty。	87
将中文字号功能提取到可以独立使用的 ctexsize。	116
解决 etoolbox 与 breqn 关于 \end 的冲突。	120
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	79
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua \LaTeX 。	52
新增 autoindent 选项。	42
新增 fontset 选项。	43
新增 linespread 选项。	42
新增 linestretch 选项。	84
新增 scheme 选项,并将 cap 和 nocap 列为过时选项。	44
新增 zhmCJK 支持选项。	43
新增 zihao 选项。	42
新增统一设置接口 \ctexset。	83
应用 \LaTeX 3 重新整理代码。	1
中文字号不再采用近似值。	117
自动检测操作系统,载入对应的字体配置。	77
\CJK@surr: 解决与 \nouppercase 的冲突。	49

v2.0.1 (2015/05/15)

General: 修复 10pt、11pt 等选项无效的问题。	45
---	----

v2.0.2 (2015/05/16)

General: 修复加载 ctex 宏包后章节标题后第一段无段首缩进的问题。	116
---	-----

v2.1 (2015/05/18 – 2015/06/19)

General: format+, nameformat+ 等带加号的选项,加号与前面的文字之间可以有可选的空格。	89
nameformat 可以接受章节名字为参数。	89
不依赖 ifpdf 宏包。	40
不再设置 hyperref 宏包的 colorlinks 选项。	77
给 pdf \LaTeX 下的非 UTF-8 编码 CJK 字体族加上 CMap。	47

将章节标题设置功能提取到可以独立使用的宏包

ctexheading 中。	1
新的标题格式选项 aftertitle。	89
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。	110
\ctex_if_platform_macos:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	121
hyperref: 补充定义 \hypersetup。	45

v2.2 (2015/06/21 – 2015/06/30)

General: before skip 和 after skip 选项的符号不再有特殊意义。	89
before skip、after skip 和 indent 选项支持表达式。	92
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	92
给 enumitem 宏包注册 \chinese 和 \zhnum。	87
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。	140
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。	110
删去 etoolbox 与 breqn 的兼容补丁。	120
新的标题格式选项 afterindent。	89
新的标题格式选项 numbering。	89
新的标题格式选项 runin。	89
新增子宏包 ctexpatch 实现给宏打补丁功能。	142

v2.3 (2015/09/17 – 2016/01/05)

General: .value_required: 和 .value_forbidden: 已过时。	42
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。	40
更新 LuaTeX-jan 支持 (20150922.0)。	52
更新 unicode-math 宏包补丁。	55
兼容 titleps 宏包。	104
修复 nameformat 作用域问题。	89
与 \LaTeX 3 (2015/12/20) 同步。	38

v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 up \LaTeX 。	74
加强 beamer 宏包支持。	1
提供 up \LaTeX 的 NFSS 字体定义。	123
提供 translator 宏包的中文定理名称翻译。	139
正确更新 CJK 包的 \CJKfamilydefault。	77
正确设置 up \LaTeX 下字体命令。	74
\ctex_parse_name:NN: 修复宏名解析错误。	143

v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。	116
beamer 不调整默认字体大小。	115
使用 bootfont.bin 判断 Windows XP 以避免权限问题。	130
随字体更新 up \LaTeX 的 \xkanjiskip。	81
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	90
正确更新 up \LaTeX 的 \CJKfamilydefault。	77
\ccwd: 正确设置 up \LaTeX 下的 \ccwd。	79
\chinese: 支持 \pagenumbering。	87
zihao: 不允许无参 zihao 选项。	42

v2.4.2	(2016/05/15)	v2.4.13	(2018/03/23)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	55	<code>\CTEX@selectfont@hook</code> : 修正导言区 <code>\selectfont</code> 钩子位置。	80
<code>\em</code> : 兼容 <code>upL^AT_EX</code> 2016/05/07u00 的定义。	74	v2.4.14	(2018/05/01)
v2.4.3	(2016/06/03 – 2016/08/26)	General: 配置 <code>macnew</code> 的默认字体设置。	128
General: 更新 <code>unicode-math</code> 补丁。	61	区分 <code>macold</code> 及 <code>macnew</code> 。	128
简化 <code>fontspec</code> 补丁。	55	为 <code>macnew</code> 配置字体命令。	131
确保 <code>\proofname</code> 非空。	87	v2.4.15	(2019/01/29 – 2019/04/05)
删除选项 <code>part/fixbeforeSKIP</code> 和 <code>chapter/fixbeforeSKIP</code> 。	90	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	92
新的标题格式选项 <code>fixSKIP</code> 。	89	将 <code>upL^AT_EX</code> 的默认字体由 <code>mc</code> 改为 <code>zhrm</code> , 并启用 <code>\jfam</code> 。	74
v2.4.4	(2016/09/09 – 2016/09/19)	将 <code>JY2</code> 和 <code>JT2</code> 编码的字体定义提取到单独的文件中。	122
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	52	局部指定 <code>autoindent</code> 为 <code>false</code> , 并交换 <code>\CTEX@XXX@indent</code> 与 <code>\CTEX@XXX@format</code> 的顺序。	107
改进 <code>hyperref</code> 宏包的标题锚点设置。	101	同步 <code>L^AT_EX 3</code> 2019/03/05。	1
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	49	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	127
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	103	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	74
提供 <code>\CTEXifname</code> 。	91	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	92
提供 <code>\partmark</code> 。	91	v2.4.16	(2019/05/11 – 2019/05/29)
新的标题格式选项 <code>break</code> 。	89	General: 更好地兼容 <code>nameref</code> 宏包。	101
v2.4.5	(2016/10/01 – 2016/10/25)	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法, 在其标题内部禁用 <code>autoindent</code> 。	92
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	90	允许设置 <code>autoindent</code> 为 0。	40
新的标题格式选项 <code>hang</code> 。	89	v2.5	(2019/10/25 – 2020/04/30)
新的标题格式选项 <code>tocline</code> 。	89	General: <code>ubuntu</code> 改用思源(Noto CJK)和文鼎字库, 不再支持使用 <code>pdfL^AT_EX</code> 编译。	129
<code>\ps@headings</code> : 修复补丁失败。	105	标题选项 <code>format</code> 也可以接受参数。	92
v2.4.6	(2016/10/31 – 2016/11/20)	不再支持 Windows XP 系统, <code>windowsold</code> 和 <code>windowsnew</code> 成为过时字库选项。	130
General: <code>\CTEXifname</code> 初始为假。	91	不再自动载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	79
支持字体属性可选项在后的新语法。	66	操作系统检测移动至载入中文字库处, 且不再需要依赖特定引擎。	77
重新初始化 <code>\ifthechapter</code> 等。	104	处理 <code>\ctex_file_input:n</code> 在 <code>ctexsize</code> 中未定义的错误。	38
v2.4.7	(2016/12/23 – 2016/12/27)	给 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下的文档类指定驱动为 <code>DVIPDFMx</code> 。	46
General: 依赖 <code>pxeverysel</code> 宏包。	75	更新 <code>LuaTeX-j</code> 支持(20200412.0)。	52
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	105	兼容 <code>titletoc</code> 宏包。	105
v2.4.8	(2017/02/23)	删除 <code>fontspec</code> 补丁。	55
General: 解决与 <code>fontspec</code> 2017/01/24 v2.5d 的字体族匹配兼容问题。	67	使用环境变量代替绝对路径查找字体。	130
v2.4.9	(2017/02/27)	所有引擎下默认编码均设为 UTF-8。	43
General: 调整 <code>unicode-math</code> 补丁的代码顺序。	61	为 <code>macnew</code> 增加粗楷体、隶书和圆体的定义。	128
v2.4.10	(2017/07/19 – 2017/07/23)	移除 <code>\CTEXunderdot</code> 、 <code>\CTEXunderline</code> 、 <code>\CTEXunderdblline</code> 、 <code>\CTEXunderwave</code> 、 <code>\CTEXsout</code> 、 <code>\CTEXxout</code> 、 <code>CTEXfilltwosides</code> 等命令和环境。	79
General: 常数 <code>\c_minus_one</code> 已过时。	38	应用新内核中的 <code>\labelformat</code> 。	110
定义 <code>\cht</code> 、 <code>\cdp</code> 和 <code>\cwd</code> 。	82	允许 <code>macnew</code> 在 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下使用。	128
使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 (<code>L^AT_EX 3</code> 2017/07/19)。	38	在 <code>ctexsize</code> 也载入 <code>fix-cm</code> 。	38
v2.4.11	(2017/08/17 – 2017/11/21)	增加宏包开头钩子。	140
General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>JAchar</code> 。	53	增加字体映射文件 <code>zhmacfonts.tex</code> 。	135
不把希腊和西里尔字母设置为 <code>JAchar</code> 。	53	重构标题选项 <code>indent</code> 和 <code>hang</code> 。	90
因上游 <code>l3keys</code> 变化, 重新定义 <code>format_u+</code> 等带空格加号的选项。	89	重构字体选项 <code>AlternateFont</code> 。	68
<code>\ps@headings</code> : 补充页眉空格。	105	<code>\CTEX@selectfont@hook</code> : 兼容 <code>KOMA-Script</code> 的 <code>\selectfont</code> 补丁。	80
v2.4.12	(2017/12/05 – 2018/01/27)	<code>\ctex_if_platform_macos:TF</code> : 改用 <code>/System/Library/Fonts/Menlo.ttc</code> 为特征文件。	121
General: 同步 <code>L^AT_EX 3</code> 2017/12/16。	1	<code>fntef</code> : 仅在该选项启用时会载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	44
修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	83		
正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	74		
<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	120		

v2.5.1	(2020/05/02)	v2.5.4	(2020/06/07 – 2020/08/16)
General: zhconv 更名为 ctex-zhconv。	1	General: 更新 LuaTeX-jā 支持(20200808.0)。	52
v2.5.2	(2020/05/05 – 2020/05/06)	兼容 L ^A T _E X 2020/10/01 的钩子机制。	140
General: ctexmakespa.tex 更名为 ctex-spa-make.tex。	136	同时兼容 cleveref 和 hyperref。	109
ctexspamacro.tex 更名为 ctex-spa-macro.tex。	136	修正主要字体命令补丁。	76
zhadobefonts.tex 等字体映射文件更名为		应用 l3cctab。	39
ctex-zhmap-*.tex。	132	v2.5.5	(2020/10/06 – 2020/10/17)
兼容 L ^A T _E X 2020-02-02 之前的版本。	76	General: 放弃应用 l3cctab。	39
新增标题选项 secnumdepth 和 tocdepth。	108	进一步应用 L ^A T _E X 2020/10/01 的新钩子。	76
修正 macnew 和 ubuntu 字库的 CJKpunct 标点信息。	127	\CJKhook: 不再通过旧的钩子命令来定义。	51
v2.5.3	(2020/05/31 – 2020/06/06)	v2.5.6	(2021/01/11 – 2021/02/16)
General: 不再依赖 xunicode, 单独补丁 tuenc.def。	53	General: 使用正确的导言区末尾钩子。	141
兼容 cleveref。	109	\f@nch@initialise: 更新 fancyhdr 宏包的补丁。	106
正确关闭和恢复 L ^A T _E X3 语法环境。	39		

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	
_	3364, 3371, 3384, 3387, 3411, 3418
\#	512, 3260, 3319
.../afterindent	24
.../aftername	20
.../aftername+	20
.../afterskip	23
.../aftertitle	21
.../aftertitle+	21
.../beforeskip	23
.../break	24
.../break+	24
.../fixskip	24
.../format	18
.../format+	18
.../hang	22
.../indent	23
.../name	17
.../nameformat	18
.../nameformat+	18
.../number	17
.../numberformat	20
.../numberformat+	20
.../titleformat	20
.../titleformat+	20
.../tocline	25
\	1720
\\	15, 33, 34, 136, 182, 184, 231, 240, 668, 1225, 1307, 2375, 3831, 3832, 3945, 3946, 4194, 4196, 4225, 5311, 5312, 5313, 5331, 5422
\{	1236, 1237, 5313, 5332
\}	1236, 1237, 5313, 5332
10pt	8, <u>362</u>
11pt	8, <u>362</u>
12pt	8, <u>362</u>
A	
\abovedisplayshortskip	4036, 4044, 4055, 4075, 4083, 4094
\abovedisplayskip	4035, 4038, 4043, 4050, 4054, 4061, 4074, 4077, 4082, 4089, 4093, 4100
\abstractname	2359
abstractname	12
\addCJKfontfeature	1307, 1351
\addCJKfontfeatures	<u>1324</u>
\addcontentsline	2587, 2632, 2636, 2641, 2673, 2677, 2682, 2782, 2787, 2793, 2799, 2928, 2942
\AddEnumerateCounter	2314, 2316, 2317, 2318
\addpenalty	2885, 3672, 3705, 3739, 3753, 3767
\addtocontents	2575, 2581, 2804, 2805
\addvspace	2576, 2582, 2593, 2596, 2804, 2805, 2885, 2889, 3434, 3445, 3453, 3464, 3472, 3483
adobefonts	32
\algorithmname	2366, 2382
algorithmname	13
AlternateFont	29, 35, <u>1436</u>
\appendix	<u>3097</u> , <u>3549</u>
appendix/name	25, <u>3086</u>
appendix/number	26, <u>3086</u>
appendix/numbering	25, <u>3086</u>
\appendixname	2361, 3816
appendixname	12
\arabic	3719
\AtBeginDocument	571, 1710
\AtBeginDvi	466
\AtBeginShipoutFirst	470, 1903, 1918
\AtBeginUTFCommand	791

\AtEndOfClass 160
 \AtEndOfPackage 161
 \AtEndUTFCommand 792
 autoindent 11, 210, 2164

B

\baselinestretch 2223
 \begin 3441, 3460, 3479
 \begingroup 436, 2646, 2688, 2708, 2729, 2828,
 2853, 2918, 2956, 2989, 3429, 3448, 3467, 4817, 4851, 4858
 \belowdisplayshortskip . 4037, 4045, 4056, 4076, 4084, 4095
 \belowdisplayskip 4038, 4050, 4061, 4077, 4089, 4100
 \bfdefault 1092, 1098, 1426, 1433
 \bfseries 2651, 2655, 2693, 2697, 2713,
 2734, 2834, 2839, 2858, 3600, 3602, 3606, 3619, 3620,
 3623, 3649, 3651, 3657, 3674, 3678, 3699, 3733, 3750, 3764
 \bibname 2364, 2367, 2375, 2383, 2389, 2390
 bibname 12

bool commands:

\bool_gset_false:N 273, 278
 \bool_gset_true:N 268
 \bool_if:NTF 86, 1265, 3011, 3201, 3211, 3222, 3835
 \bool_lazy_or:nnTF 4171, 4802, 5450
 \bool_new:N 156, 1302
 \bool_set_false:N 1277
 \bool_set_true:N 1295
 \c_false_bool 3045, 3047
 \c_true_bool 3044, 3046

box commands:

\box_ht:N 4116
 \box_new:N 62
 \box_use_drop:N 3021
 \box_wd:N 1951, 3020, 3315

C

c5size 32
 cap 33, 321
 captiondelimiter 34, 4123
 \catcode 4817, 4818, 4819, 4820, 4821, 4825, 4826
 CCT 32
 CCTfont 32
 \ccwd 23, 27, 154, 214, 1947, 1992,
 1993, 1994, 2170, 2208, 2232, 2233, 2240, 2266, 2267,
 2268, 2271, 2273, 2274, 2285, 2296, 2298, 4133, 4134, 4135
 \cdp 2078
 \centering 2689, 2730, 3430,
 3449, 3468, 3606, 3618, 3623, 3627, 3657, 3678, 3686, 3721
 \chapter 2759, 3841
 chapter/beforeskip 34
 chapter/fixbeforeskip 34
 chapter/lofskip 25, 2502
 chapter/lotskip 25, 2502
 chapter/numbering 16
 chapter/pagestyle 21, 2502
 \chaptermark 2803
 \chaptername 3646

char commands:

\char_set_catcode:nn 107, 108
 \char_set_catcode_letter:n . 95, 1713, 3335, 3556, 5414
 \char_set_catcode_math_superscript:n 96
 \char_set_catcode_other:N 512, 1004, 3260, 3319
 \char_set_catcode_other:n 5438
 \char_set_catcode_space:n 797, 2083
 \char_value_catcode:n 90, 91

CharRange 29, 1436

\Chinese 33, 2310
 \chinese 28, 2308, 2318, 3593, 3656
 \cht 2078

\CJKbold
 4271, 4272, 4282, 4283, 4301, 4302, 4304, 4305, 4307,
 4308, 4312, 4313, 4315, 4316, 4318, 4319, 4335, 4336,
 4338, 4339, 4341, 4342, 4346, 4347, 4349, 4350, 4352, 4353

\CJKfamily 1164, 1184, 1531, 1728, 1834, 1835, 1836,
 1837, 4792, 4793, 4794, 4795, 4797, 4798, 4800, 4810, 4811

\CJKfamilydefault 586, 1390, 1392, 1396, 1403,
 1416, 1828, 1829, 1837, 1846, 1853, 1857, 1865, 1876, 1878

\CJKfontspec 1324

\CJKglue 1950, 1964, 1978, 1990

\CJKhook 599

\CJKnormal 4264, 4265, 4266, 4267, 4268, 4269, 4270,
 4275, 4276, 4277, 4278, 4279, 4280, 4281, 4300, 4303,
 4306, 4311, 4314, 4317, 4334, 4337, 4340, 4345, 4348, 4351

\CJKpunctmapfamily
 . 634, 635, 636, 637, 638, 639, 643, 644, 645, 646, 647, 648

\CJKrmdefault 535, 1232, 1243,
 1355, 1392, 1395, 1396, 1829, 1831, 1834, 1840, 1843,
 1861, 4388, 4391, 4392, 4397, 4398, 4399, 4452, 4455,
 4457, 4462, 4463, 4464, 4518, 4521, 4522, 4529, 4530,
 4531, 4598, 4601, 4602, 4609, 4611, 4612, 4679, 4682,
 4685, 4691, 4692, 4693, 4744, 4745, 4746, 4754, 4755, 4756

\CJKsfdefault 536, 1233,
 1244, 1362, 1832, 1835, 1841, 1844, 1862, 4389, 4453,
 4456, 4519, 4523, 4599, 4603, 4610, 4680, 4683, 4747, 4748

\CJKtilde 587

\CJKttdefault 537, 1234, 1245, 1368, 1833, 1836,
 1842, 1845, 1863, 4390, 4454, 4520, 4600, 4681, 4685, 4749

clearalternatefont 30, 1615

\cleardoublepage 2607, 2760, 3616, 3644

\clearpage 2609, 2760, 3616, 3644

clist commands:

\clist_clear:N 1113, 5131
 \clist_concat:NNN 1133
 \clist_const:Nn 613
 \clist_gput_right:Nn 378, 390, 411, 412
 \clist_gset:Nn 1378
 \clist_if_empty:NTF 1461
 \clist_map_break:n 3180, 3826
 \clist_map_function:nN . 1444, 1618, 1620, 1653, 1667
 \clist_map_inline:Nn 603, 1561
 \clist_map_inline:nn
 363, 626, 1660, 1942, 3175, 3823, 3963, 3998, 5120
 \clist_new:N 163, 1130, 1379, 1457, 1504, 5149, 5306

- \clist_put_left:Nn 1134, 3892, 3918
- \clist_put_right:Nn 1294, 1550, 5143
- \clist_set:Nn 1117, 1447, 1487, 1488, 1707, 5307
- \clist_use:Nn 5316
- \clubpenalty 2988, 2995
- \contentsname 2354
- contentsname 12
- continuation 13
- cs commands:
 - \cs_generate_variant:Nn
 - ... 125, 488, 500, 1079, 1196, 1283, 1303, 1304, 1505, 1521, 1593, 1614, 1653, 1661, 3023, 3505, 5376, 5505, 5612
 - \cs_gset_eq:NN 269, 274, 279, 471, 520, 910, 1270, 1273, 3133, 3134, 3139, 3164
 - \cs_gset_nopar:Npx 905
 - \cs_gset_protected:Npn 3098, 3145, 3160
 - \cs_gset_protected:Npx 1161, 1525, 1611, 3074
 - \cs_gset_protected_nopar:Npx 947
 - \cs_if_eq:NNTF . 1038, 3131, 3163, 5387, 5393, 5538, 5618
 - \cs_if_exist:NTF
 - 44, 354, 462, 497, 899, 915, 945, 979, 1057, 1524, 1609, 1807, 1819, 2042, 3246, 3525, 3563, 3841, 3843
 - \cs_if_exist_p:N 5451, 5452
 - \cs_if_exist_use:N 1180, 1181
 - \cs_if_exist_use:NTF 4, 747, 882, 1202, 1629, 5383
 - \cs_if_free:NTF 482, 492, 605, 621, 1267, 1425, 2009, 2012, 2183, 2197, 2314, 3194, 3239, 3285, 3290, 3294
 - \cs_new:Npn 146, 153, 964, 994, 999, 1001, 1005, 1109, 1175, 1183, 1228, 1239, 1682, 2308, 2484, 3048, 3056, 3064, 3083, 3219, 3283, 3288, 5150, 5495, 5613
 - \cs_new:Npx 4142, 4148
 - \cs_new_eq:NN 6, 7, 8, 9, 20, 23, 140, 141, 142, 158, 159, 355, 489, 590, 652, 924, 949, 1015, 1045, 1351, 1358, 1872, 1990, 2000, 2006, 2309, 2310, 2311, 2542, 2561, 2564, 3032, 3042, 3043, 3044, 3045, 3046, 3047, 3097, 3121, 3122, 3566, 3839, 3844, 4806, 4807, 5125, 5441, 5442, 5542, 5614
 - \cs_new_protected:Npn
 - 65, 68, 70, 72, 75, 81, 99, 104, 114, 144, 160, 161, 169, 171, 176, 458, 460, 465, 480, 490, 495, 501, 533, 542, 551, 591, 597, 601, 619, 624, 632, 641, 650, 719, 721, 727, 729, 737, 739, 744, 757, 762, 777, 879, 889, 902, 907, 913, 966, 977, 989, 1008, 1017, 1033, 1047, 1072, 1110, 1131, 1140, 1159, 1176, 1186, 1211, 1249, 1261, 1263, 1285, 1287, 1310, 1386, 1411, 1420, 1443, 1445, 1452, 1459, 1465, 1472, 1474, 1479, 1485, 1506, 1515, 1522, 1528, 1539, 1547, 1553, 1559, 1577, 1586, 1594, 1606, 1624, 1641, 1659, 1662, 1664, 1677, 1696, 1704, 1718, 1721, 1723, 1747, 1761, 1774, 1787, 1802, 1809, 1816, 1851, 1883, 1887, 1925, 1933, 1937, 1947, 1961, 1991, 1995, 2007, 2057, 2182, 2184, 2209, 2217, 2229, 2248, 2258, 2263, 2417, 2447, 2495, 2525, 2532, 2537, 2545, 2552, 2559, 2565, 2567, 2570, 2586, 3000, 3006, 3008, 3016, 3024, 3033, 3114, 3124, 3156, 3167, 3187, 3192, 3197, 3207, 3217, 3261, 3309, 3327, 3499, 3521, 3530, 3536, 3541, 3551, 3936, 3951, 3983, 3988, 4018, 4024, 4138, 4154, 4164, 4166, 5117, 5126, 5141, 5321, 5327, 5334, 5344, 5346, 5348, 5353, 5355, 5357, 5359, 5369, 5371, 5377, 5382, 5385, 5391, 5398, 5403, 5408, 5419, 5423, 5428, 5433, 5443, 5448, 5467, 5518, 5533, 5543, 5555, 5564, 5583, 5588, 5600, 5615
 - \cs_new_protected:Npx 505, 750, 1334, 3248, 5342
 - \cs_new_protected_nopar:Npn 925, 939
 - \cs_replacement_spec:N 5459, 5481
 - \cs_set:Npn 2003, 3300
 - \cs_set_eq:NN 143, 215, 220, 225, 549, 569, 594, 598, 607, 622, 731, 764, 772, 779, 781, 782, 918, 919, 1016, 1046, 1064, 1107, 1164, 1524, 1531, 1541, 1542, 1543, 1609, 1610, 1720, 1856, 2021, 2024, 2025, 2028, 2325, 2330, 2333, 2338, 2566, 2568, 3035, 3036, 3037, 3038, 3039, 3040, 3128, 3171, 3181
 - \cs_set_nopar:Npn 1746
 - \cs_set_nopar:Npx 3209, 5361
 - \cs_set_protected:Npn
 - 593, 1050, 1065, 1964, 3241, 3298, 4025, 5446, 5510
 - \cs_set_protected:Npx 742, 1997, 5435, 5436
 - \cs_to_str:N 60, 1333, 1805, 1812, 5444
 - \cs_undefine:N . 1144, 1145, 1149, 1150, 1544, 2029, 5383
- cs4size 32
- \csname 437, 440, 441, 2886, 2895, 2896, 2897, 2904, 2924, 2925, 2927, 2930, 2939, 2940, 2941, 2944, 3558, 3559, 4850, 4852, 4853, 4854, 4859, 4863
- \CTeX 29, 4119
- ctex commands:
 - \ctex_add_cmap:n 485, 490
 - \ctex_add_cmap:Nn 493, 495, 500
 - \ctex_add_dvi_zhmap: 465, 467, 471
 - \ctex_add_to_selectfont:n 1995, 2048, 2076, 2103, 2196
 - \ctex_after_end_preamble:n ... 40, 143, 467, 5346, 5353
 - \ctex_appto_cmd:NnnTF 516, 769, 1712, 5428
 - \ctex_assign_heading_name:nn 2453, 2492, 3090
 - \ctex_at_begin_package:nn 142, 5321
 - \ctex_at_end:n .. 160, 292, 330, 336, 347, 352, 1990, 4232
 - \ctex_at_end_package:nn 142, 468, 519, 788, 794, 1055, 1897, 1899, 1917, 1922, 1929, 2312, 3143, 3158, 3227, 3234, 3293, 3295, 3296, 3332, 3375, 3534, 3572, 4121, 5334
 - \ctex_at_end_preamble:n 357, 576, 583, 1850, 5344, 5353
 - \ctex_auto_ignorespaces: 590, 598, 2113
 - \ctex_CJK_input:n 542
 - \ctex_cleveref_hook: 3534, 3541
 - \ctex_declare_math_sizes:nnnn 3983, 4017
 - \ctex_define:n
 - 64, 1615, 1654, 2105, 2126, 2164, 2198, 2326, 2352, 2380, 2390, 2393, 2396, 2524, 3086, 3490, 4123, 4199
 - \ctex_define_option:n 64, 191, 384, 387
 - \ctex_deprecated_command:Nn
 - 169, 2147, 2149, 2157, 2159, 2295, 2302
 - \ctex_deprecated_option:n
 - 169, 173, 229, 238, 329, 335, 339, 346, 351, 4127
 - \ctex_deprecated_option:nn 169, 296
 - \ctex_detect_platform: 4154, 4169, 4184
 - \ctex_family_cmap:nn 480, 489

`\ctex_file_input:n` 75,
 393, 538, 653, 780, 2139, 3850, 3885, 3890, 3895, 3896,
 3911, 3916, 3922, 3923, 4028, 4029, 4191, 4232, 4569, 4570
`\ctex_fix_varioref_label:n` 3533, 3536
`\ctex_fontset_case:nnn` 4142, 4503, 4726
`\ctex_fontset_case:nnnn` . 4148, 4364, 4428, 4573, 4662
`\ctex_fontset_error:n` 4138, 4365,
 4401, 4429, 4466, 4533, 4574, 4576, 4577, 4614, 4663, 4695
`\ctex_gadd_hook:Nn` 5369
`\ctex_gadd_ltxhook:nn`
 462, 463, 1807, 1812, 5342, 5345, 5347, 5349
`\ctex_gadd_package_hook:nnn` ... 5325, 5338, 5348, 5369
`\ctex_get_macro_meaning:N` 5513
`\ctex_get_macro_meaning:NTF` 5509, 5535, 5591
`\ctex_hang_from:n` 3008
`\ctex_heading_depth:nn` 3493, 3496, 3499
`\ctex_heading_glue:n` 3024
`\ctex_heading_hang:Nnn` 3002, 3007, 3008
`\ctex_hypersetup:n`
 78, 355, 1880, 1892, 1895, 1910, 1913, 1916
`\ctex_if_autoindent_touched:TF` 158, 215, 220, 225, 3877
`\ctex_if_ccglue_touched:TF` 1975, 2219
`\ctex_if_ccglue_touched_p:` 1975
`\ctex_if_platform_macos:TF` 4159, 4164
`\ctex_if_preamble:TF` 142, 4203
`\ctex_if_rescanable:NnTF` 5533, 5567
`\ctex_ignorespaces_case:N` 591, 2110, 2113, 2116
`\ctex_indent_box:n` 2532
`\ctex_load_fontset:` 4166, 4211, 4231
`\ctex_load_zhmap:nnnn`
 533, 4395, 4460, 4527, 4607, 4689, 4752
`\ctex_ltj_add_font_features:n` 1285, 1348
`\ctex_ltj_add_font_features:nn` 1285, 1286
`\ctex_ltj_add_kyenc:n` 806, 1088
`\ctex_ltj_char_range_key:nn` 1701, 1704
`\ctex_ltj_clear_alt_font:n` 843, 1572
`\ctex_ltj_clear_alterate_font:n` 1618, 1624
`\ctex_ltj_declare_alterate_shape:nnnnnn` 1533, 1553
`\ctex_ltj_declare_char_range:n` ... 1656, 1659, 1661
`\ctex_ltj_declare_char_range:nn` 1663, 1664
`\ctex_ltj_def_char_range_key:n` 1669, 1696
`\ctex_ltj_ensure_default_family:` 1386, 1869
`\ctex_ltj_extract_font:` 918, 925
`\ctex_ltj_family_if_exist:nNTF`
 1188, 1197, 1390, 1395, 1413, 1416, 1627
`\l_ctex_ltj_family_tl`
 1190, 1195, 1252, 1253, 1268, 1272, 1275, 1286, 1621, 1622
`\ctex_ltj_fontspec:nn` 1249, 1261, 1296, 1343
`\ctex_ltj_get_and_define_fonts:nN` 1008
`\ctex_ltj_get_and_define_fonts_al:nN` ... 1012, 1015
`\ctex_ltj_get_and_define_fonts_ja:nN` ... 1011, 1017
`\ctex_ltj_if_alterate_shape_exist:nTF`
 928, 941, 959, 968
`\ctex_ltj_if_jfont:nTF` 62, 994, 1006, 1010
`\ctex_ltj_if_jfont_math:NTF` 1001, 1040
`\ctex_ltj_is_kenc:n` 809, 996
`\ctex_ltj_math_group_hook:` ... 61, 61, 1039, 1045, 1065
`\ctex_ltj_patch_external_font:n` 812, 965
`\ctex_ltj_pickup_alt_font:nn` 849, 970
`\ctex_ltj_pickup_font:` . 895, 913, 984, 1020, 1022, 1024
`\ctex_ltj_reset_alterate_font:n` 1620, 1641
`\ctex_ltj_save_alterate_family:Nnnn` 1498, 1519, 1522
`\ctex_ltj_save_alterate_seq:Nn` 1469, 1474
`\ctex_ltj_save_char_range:n` 1667, 1674
`\ctex_ltj_save_char_range:nn` 1676, 1677
`\ctex_ltj_select_alterate_font:` 966, 2051
`\ctex_ltj_select_font:` 879, 2050
`\ctex_ltj_set_alt_font:nnnn` 834, 1588, 1591
`\ctex_ltj_set_alterate_family:nn` 1473, 1506
`\ctex_ltj_set_alterate_family:nnn` . 1518, 1526, 1528
`\ctex_ltj_set_alterate_family:Nnnnn` 1510, 1515, 1521
`\ctex_ltj_set_alterate_family:nnTF` 1118, 1459
`\ctex_ltj_set_alterate_prop:n` 1438, 1443
`\ctex_ltj_set_alterate_shape:n` 1579, 1586, 1593, 1598
`\ctex_ltj_set_alterate_shape:nnN` 1565, 1594
`\ctex_ltj_set_alterate_shape:nnn` 1568, 1574
`\ctex_ltj_set_alterate_shape:nnnn` 1576, 1577
`\ctex_ltj_set_alterate_shape:Nnnnnnn` .. 1556, 1559
`\ctex_ltj_set_family:nnn`
 1110, 1280, 1327, 1337, 1355, 1362, 1368, 1374
`\ctex_ltj_set_kanjiskip:N` 1721, 1969
`\ctex_ltj_set_math_letter:NN` 1047, 1059, 1067
`\ctex_ltj_set_xkanjiskip:N` 1721, 2064
`\ctex_ltj_subst_font:` 919, 939
`\ctex_ltj_swap_cs:NN` 1532, 1539
`\ctex_ltj_switch_family:n`
 1185, 1186, 1196, 1253, 1281, 1335
`\ctex_ltj_update_mathfont:` 1408, 1411
`\ctex_ltj_update_mathfont:n` 1414, 1417, 1420
`\ctex_ltj_use_jfont:` 829, 935
`\ctex_ltj_use_math_group:Nn` 1033, 1064
`\ctex_ltj_zero_globaldefs:` 724, 872
`\ctex_make_spa:nn` 5117
`\ctex_mono_jfm:n` 1072, 1080, 2133
`\ctex_parse_name:NN` 1821, 2032, 5439, 5443
`\ctex_parse_name:NNn` 5444, 5448, 5505
`\ctex_patch_boot:NNnnTF`
 5400, 5405, 5410, 5425, 5430, 5433
`\ctex_patch_cmd:Nnn` 76, 1742, 3263, 3269,
 3275, 3347, 3350, 3354, 3360, 3370, 3377, 3380, 3383,
 3386, 3394, 3397, 3401, 3407, 3417, 4133, 4134, 4135, 5408
`\ctex_patch_cmd_all:NnnnTF` 3553, 5403
`\ctex_patch_cmd_once:NnnnTF` 1736, 2034, 5398
`\ctex_patch_failure:N`
 .. 522, 773, 775, 1716, 2039, 3326, 3338, 3561, 5417, 5419
`\ctex_plane_to_utfxvibe:Nn` 550
`\ctex_pop_file:` 79, 99, 123, 547
`\ctex_preto_cmd:NnnTF` 765, 3323, 3334, 5423
`\ctex_provide_font_hook:NNN` 1802
`\ctex_punct_map_bfseries:nn` 624,
 4391, 4398, 4455, 4456, 4463, 4522, 4523, 4524, 4530,
 4602, 4603, 4604, 4611, 4683, 4684, 4692, 4745, 4748, 4755

- \ctex_punct_map_family:nn 619, 4388, 4389, 4390, 4397, 4452, 4453, 4454, 4462, 4518, 4519, 4520, 4529, 4598, 4599, 4600, 4609, 4610, 4679, 4680, 4681, 4691, 4744, 4747, 4749, 4754
- \ctex_punct_map_itshape:nn 641, 4392, 4399, 4457, 4464, 4521, 4531, 4601, 4612, 4682, 4693, 4746, 4756
- \ctex_punct_map_series:nnn 628, 629, 632
- \ctex_punct_set:n 601, 4387, 4396, 4451, 4461, 4517, 4528, 4597, 4608, 4678, 4690, 4730, 4753
- \ctex_punct_space:nn 650
- \ctex_push_file: 77, 81, 116, 544
- \ctex_scheme_input:n 113, 3574, 3852
- \ctex_select_size: .. 2171, 2178, 2181, 2182, 2203, 2261
- \ctex_set:n 64, 293, 2141, 2161, 3876, 3878
- \ctex_set:nn 31, 64, 174, 2151, 3577, 3635, 3663, 3695, 3730, 3747, 3761, 3777, 3785, 3793, 3801, 3808, 3809, 3810, 4239
- \ctex_set_default_ccwd:Nn 144, 224, 2177, 2202
- \ctex_set_deprecated_option:n 169, 203, 204, 258, 259, 260, 282, 287, 321, 322
- \ctex_set_font_size:Nnn 4018, 4033, 4041, 4052, 4063, 4064, 4065, 4066, 4067, 4068, 4069, 4072, 4080, 4091, 4102, 4103, 4104, 4105, 4106, 4107, 4108
- \ctex_set_ignorespaces: 591, 600
- \ctex_set_upfamily:nnn ... 1747, 1794, 4411, 4412, 4413, 4414, 4476, 4477, 4478, 4479, 4543, 4544, 4545, 4546, 4547, 4548, 4623, 4624, 4625, 4626, 4627, 4628, 4705, 4706, 4707, 4768, 4769, 4770, 4771, 4772, 4773, 4774
- \ctex_set_upfonts:nnnnnn 1787, 1797, 4404, 4469, 4536, 4761
- \ctex_set_upmap:nnn .. 1761, 1789, 1790, 1791, 1792, 1795, 4549, 4550, 4617, 4618, 4619, 4620, 4704, 4775, 4776
- \ctex_set_upmap_unicode:nnn 1774, 1796, 4621, 4622, 4698, 4700, 4702
- \ctex_set_zhmap:n 457, 538, 1763, 1776
- \ctex_titleps_hook: 3259, 3294, 3295
- \ctex_titlesec_hook: 3167, 3231
- \ctex_update_ccglue: 1961, 2226, 2246, 2289
- \ctex_update_ccwd: 1947, 2220, 2297
- \ctex_update_default_family: 50, 576, 1850, 1851
- \ctex_update_em_unit: 1991, 2211, 2265
- \ctex_update_kanjisize: 2078
- \ctex_update_parindent: 2189, 2248, 2290
- \ctex_update_size: 2183, 2184
- \ctex_update_stretch: 2188, 2209
- \ctex_update_xkanjiskip: 2056, 2076
- \ctex_update_ziju: 2191, 2263
- \ctex_use_package_hook:nn 5377, 5388, 5394
- \ctex_use_zhmap: 460, 463, 466, 470
- \ctex_varioref_hook: 3530, 3564, 3572
- \ctex_zhmap_case:nnn 159, 269, 274, 279, 4367, 4431, 4505, 4581, 4665, 4728
- \ctex_zihao:n 3935, 3936
- \ctex_ziju:n 2257, 2258
- ctex internal commands:
 - \g_ctex_after_end_preamble_hook_tl 5356, 5360, 5368
 - _ctex_assign_heading_name:nnn 2492
 - \l_ctex_autoindent_tl 157, 214, 219, 224, 2170, 2174, 2177, 2250, 2253, 2528
 - _ctex_calc_bounds:nn 5145, 5146, 5150
 - _ctex_ccglue: 1978, 1990
 - \l_ctex_ccglue_skip 1965, 1969, 1972, 1974, 1982, 1986, 2224, 2236, 2245, 2280, 2288
 - \c_ctex_class_tl 117, 119, 415, 420, 425, 430, 3839, 3845, 3847, 3848, 3850
 - _ctex_clear_fntef_color:n ... 1925, 1933, 1937, 1944
 - _ctex_cleveref_hook_aux:N 3541
 - \c_ctex_cmap_encoding_seq 478, 484
 - \l_ctex_ctexcap_options_clist 5306, 5307, 5316
 - _ctex_def_heading_keys:n 2447, 2522
 - _ctex_default_ccwd_aux:n 145, 146
 - _ctex_default_ccwd_aux:w 149, 153
 - \g_ctex_encoding_tl 155, 248, 251, 254, 401, 405, 526, 529, 578, 586, 1893, 2306, 3880, 3907, 3921
 - \g_ctex_end_preamble_hook_tl 5354, 5358, 5367
 - \c_ctex_engine_file_str 43, 2139
 - \c_ctex_engine_str 43
 - \l_ctex_family_default_init_tl 1853, 1871
 - _ctex_family_default_wrap:n 1856, 1872, 1875
 - \g_ctex_file_status_seq 83, 101, 112
 - \l_ctex_file_status_tl 101, 102, 111
 - \g_ctex_font_size_int 186, 197, 198, 199, 373, 377, 409, 3897, 3898, 3924, 3925, 4026
 - \c_ctex_font_size_prop 3938, 3949, 3993, 4020
 - \c_ctex_font_size_seq 3947, 3950, 3961
 - \l_ctex_font_size_tl 3938, 3939, 3985, 3986, 4020, 4021
 - \g_ctex_fontset_tl 257, 4157, 4160, 4161, 4168, 4172, 4173, 4176, 4177, 4180, 4186, 4191, 4208, 4210, 4215, 4230
 - _ctex_get_font_sizes:Nn 3985, 3988
 - _ctex_get_macro_meaning:w 5509
 - \l_ctex_heading_bool 298, 3835
 - \c_ctex_heading_level_prop 3501, 3506
 - \l_ctex_heading_skip 2549, 2556, 2558, 2560, 2561, 2572, 2573, 2576, 2578, 2579, 2582, 3027, 3028, 3029
 - \c_ctex_headings_cs_seq 3116, 3129, 3138, 3161
 - \c_ctex_headings_seq ... 2407, 2519, 3244, 3307, 3532
 - _ctex_hookto_cmd:Nnnnw 5425, 5430, 5588
 - _ctex_hookto_cmd_parameter:Nnnnw 5595, 5615
 - _ctex_hookto_cmd_parameterless:Nnnnw . 5594, 5600
 - _ctex_initial_heading:n 2417, 2521
 - _ctex_insert_indent: 2535, 2537, 3014
 - _ctex_left_hook_aux:nn 5613
 - \l_ctex_line_spread_fp 119, 207, 3899, 3900, 3926, 3927, 4112, 4114
 - \l_ctex_line_stretch_tl 2202, 2207, 2208, 2212
 - \l_ctex_ltj_add_alternate_bool 1265, 1277, 1295, 1302
 - _ctex_ltj_alternate_cs:n 1145, 1148, 1149, 1150, 1180, 1181, 1183, 1268, 1271, 1272, 1274, 1275, 1469, 1511, 1571, 1590, 1630, 1647
 - \l_ctex_ltj_alternate_family_tl 1535, 1538
 - \l_ctex_ltj_alternate_options_clist 1488, 1489, 1500, 1504

`\l_ctex_ltj_alternate_prop` 1114, 1454, 1455, 1458, 1468, 1476
`\l_ctex_ltj_base_CJKfamily_tl` 1115, 1129, 1178, 1571, 1590
`\l_ctex_ltj_base_family_tl` 71, 1146, 1173, 1179, 1517, 1557, 1610, 1627, 1633, 1634, 1646
`\l_ctex_ltj_char_range_clist` 1113, 1440, 1461, 1487, 1499, 1512, 1534, 1556, 1707
`\g_ctex_ltj_char_range_prop` 1563, 1664
`\l_ctex_ltj_char_range_tl` 1563, 1566, 1666, 1668, 1670, 1672, 1679
`_ctex_ltj_chardef_text_cmd:` 733, 737
`_ctex_ltj_chardef_text_cmd_aux:N` 738, 739
`_ctex_ltj_check_family:n` 1140, 1467
`\l_ctex_ltj_cmd_tl` 732, 736, 738
`\l_ctex_ltj_current_font_tl` 57, 882, 886, 887, 899, 945, 950
`\l_ctex_ltj_current_shape_tl` 968, 971, 974, 975
`_ctex_ltj_declare_char_range:nn` 1660, 1662
`\g_ctex_ltj_default_features_clist` 1133, 1378, 1379
`_ctex_ltj_family_csname:n` ... 1144, 1161, 1175, 1202
`\g_ctex_ltj_family_font_name_prop` 1120, 1137, 1142, 1213, 1289, 1388, 1400
`\g_ctex_ltj_family_font_options_prop` 1121, 1137, 1292
`\g_ctex_ltj_family_int` . 1255, 1257, 1549, 1550, 1552
`\g_ctex_ltj_family_name_prop` 1137, 1146, 1167, 1199, 1517
`_ctex_ltj_family_unknown_warning:n` 1193, 1211, 1638
`\l_ctex_ltj_font_options_clist` 1117, 1122, 1123, 1124, 1130, 1166, 1293, 1294, 1297, 1508, 1509, 1513
`\g_ctex_ltj_fontname_seq` 904, 909, 912
`_ctex_ltj_fontspec:nnn` 1256, 1263, 1283
`\g_ctex_ltj_fontspec_family_tl` 1165, 1168, 1174, 1179, 1204
`\g_ctex_ltj_fontspec_prop` 1251, 1279, 1284
`_ctex_ltj_gset_family_cs:nn` 1125, 1159
`_ctex_ltj_if_alt_set:nTF` 864, 961
`_ctex_ltj_if_jfont_math:w` 1002, 1005
`\l_ctex_ltj_jfm_tl` 1072, 1091, 1093, 1135
`_ctex_ltj_ltj_if_jfont:w` 996, 999
`\c_ctex_ltj_math_fam_int` 1099, 1100
`\c_ctex_ltj_math_family_tl` 1422, 1424, 1426, 1429, 1433
`\c_ctex_ltj_math_tl` 1094, 1095, 1097, 1099, 1374, 1413, 1423, 1428, 1432
`_ctex_ltj_msg_def_family_map:n` 1226, 1228
`_ctex_ltj_msg_family_map:n` 1158, 1225, 1239
`_ctex_ltj_noembed_wrap:n` 1107, 1109
`_ctex_ltj_pass_args:nnnn` 1310, 1326, 1336, 1342, 1354, 1361, 1367, 1373
`_ctex_ltj_patch_external_font:w` 931, 964
`_ctex_ltj_patch_text_cmd:NN` 759, 762
`_ctex_ltj_pickup_alt_font:Nn` 857, 992
`_ctex_ltj_pop_fontname:` 898, 907, 986
`_ctex_ltj_post_arg:w` 1313, 1319
`_ctex_ltj_push_alternate_prop:n` 1444, 1445
`_ctex_ltj_push_alternate_prop:nn` . 1449, 1452, 1708
`_ctex_ltj_push_fontname:n` 894, 902, 983
`\c_ctex_ltj_range_max_int` 1691, 1695
`\c_ctex_ltj_range_min_int` 1690, 1694
`_ctex_ltj_range_normalization:nn` . 1581, 1680, 1682
`\g_ctex_ltj_reset_alternate_prop` 1151, 1632, 1644, 1652
`_ctex_ltj_save_alternate_auxi:w` 1477, 1479
`_ctex_ltj_save_alternate_auxii:w` . 1482, 1483, 1485
`_ctex_ltj_save_alternate_shape:Nn` 1570, 1589, 1606
`_ctex_ltj_select_font_aux:` 883, 889
`_ctex_ltj_set_alternate_family:n` 1169, 1176
`_ctex_ltj_set_alternate_family_aux:nnn` 1463, 1472
`_ctex_ltj_set_family_aux:nnn` 1462, 1465
`_ctex_ltj_tmp:w` 1541, 1543, 1544
`\l_ctex_ltj_tmp_clist` 1447, 1448, 1450, 1457
`\l_ctex_ltj_tmp_tl` 909, 910, 1116, 1117, 1142, 1153, 1156, 1290, 1298, 1332, 1335, 1337, 1390, 1395, 1413, 1414, 1416, 1417, 1491, 1493, 1494, 1501
`\g_ctex_ltj_unknown_family_seq` ... 1215, 1217, 1222
`_ctex_ltj_update_family_uid:N` ... 1123, 1508, 1546
`_ctex_ltj_use_global_options:N` 1124, 1131, 1489, 1509
`\c_ctex_msyh_suffix_tl` 4718, 4733, 4734, 4740, 4741, 4765, 4766
`_ctex_package_loaded_warning:nn` 5324, 5327
`\l_ctex_parameter_str` 5506, 5525, 5546, 5593
`\l_ctex_parameter_tl` 5546, 5551, 5562
`_ctex_parse_name:nNNnN` 5458, 5467
`_ctex_parse_name:w` 5480, 5486, 5495
`_ctex_patch_cmd:Nnnnw` 5400, 5405, 5410, 5564
`_ctex_patch_false:w` 5436, 5442, 5579, 5581, 5598, 5624
`_ctex_patch_rebuild:Nn` 5537, 5543, 5576, 5617, 5621
`_ctex_patch_replace:nnnTF` 5571, 5583
`_ctex_patch_rescan:NNn` 5545, 5546, 5547, 5555
`_ctex_patch_toc_width:n` 3318
`_ctex_patch_true:w` 5435, 5441, 5577, 5610, 5622
`_ctex_plus_key_aux:nn` 2474, 2476, 2477, 2478, 2479, 2480, 2481, 2484
`_ctex_pop_file_aux:nnnn` 102, 104
`\l_ctex_prefix_str` 5506, 5524, 5545, 5602, 5603
`\l_ctex_prefix_tl` 5545, 5550, 5561
`_ctex_provide_font_hook_aux:NNNN` . 1804, 1809, 1816
`\l_ctex_punct_bounds_clist` ... 5131, 5137, 5143, 5149
`\c_ctex_punct_family_clist` 603, 613
`\l_ctex_punct_font` 5129, 5130
`\c_ctex_punct_seq` 5109, 5132
`\l_ctex_punct_tl` 284, 573, 574, 660, 1080, 2130, 2131, 2132, 2133
`_ctex_rebuild_cmd:w` ... 5537, 5538, 5542, 5617, 5618
`\l_ctex_replacement_str` 5506, 5526, 5547, 5585, 5586, 5620
`\l_ctex_replacement_tl` 5547, 5552, 5563
`_ctex_restore_selectfont:` 2007, 2029, 2038
`_ctex_right_hook_aux:nn` 5614

\dim_set:Nn	1951, 1992, 1993, 1994, 2212, 2222, 2253, 2266, 2270, 2298, 2534, 2555, 3010, 3312, 4116	3076, 3078, 3079, 3080, 3081, 3202, 3203, 3212, 3213, 3250, 3252, 3253, 3539, 3955, 4186, 5460, 5461, 5462, 5550	
\dim_sub:Nn	2274	\exp_not:n	148, 179, 1164, 1165, 1166, 1297, 1298, 1499, 1500, 1501, 1526, 1582, 1583, 1611, 1856, 1876, 1998, 2473, 3221, 3527, 5308, 5364, 5435, 5436, 5478, 5550, 5551, 5552, 5607, 5608
\dim_to_decimal:n	3957, 3958, 3995	\expandafter	437, 438, 440, 441, 449, 453, 3559, 4825, 4826, 4850, 4851, 4852, 4853, 4854, 4859, 4863, 4864
\dim_to_decimal_in_unit:nn	5156	\ExplSyntaxOff	88, 94, 516, 680, 766, 770, 1737, 2035, 3323, 3427, 3555, 5162, 5413
\dim_use:N	150, 4116	\ExplSyntaxOn	87, 717, 3488, 5102
\dim_zero:N	2304		
\c_max_dim	2213, 2548		
\c_zero_dim	2225, 2238, 2252, 2268, 2275, 2462, 2539, 3028, 3808		
\directlua	702, 706		
\do	448		
\document	5361, 5364		
	E		F
\edef	4843, 4844	\familydefault	1859
\else	438, 441, 442, 446, 1738, 1739, 1743, 1744, 2599, 2608, 2617, 2633, 2638, 2674, 2679, 2760, 2768, 2784, 2790, 2796, 2809, 2819, 2879, 2884, 2903, 2908, 2929, 2934, 2943, 2965, 2981, 2994, 3616, 3644, 4855, 4856, 4882, 4906, 4955, 4987, 5019, 5068	fancyhdr	33, 344
else commands:		\fangsong	4791
\else:	528, 997, 1979, 3137, 3353, 3400	\fi	438, 441, 442, 450, 451, 452, 454, 2590, 2595, 2601, 2610, 2619, 2622, 2637, 2642, 2652, 2663, 2678, 2683, 2694, 2720, 2744, 2752, 2753, 2756, 2760, 2770, 2788, 2794, 2800, 2812, 2822, 2826, 2835, 2836, 2846, 2851, 2865, 2869, 2875, 2881, 2888, 2890, 2912, 2913, 2931, 2945, 2948, 2969, 2978, 2997, 2998, 3348, 3351, 3355, 3365, 3366, 3371, 3378, 3381, 3384, 3387, 3395, 3398, 3402, 3412, 3413, 3418, 3440, 3459, 3478, 3616, 3644, 4855, 4856, 4865, 4901, 4925, 4982, 5014, 5038, 5095
\em	1736	fi commands:	
\eminnershape	1738, 1739, 1744, 1746	\fi:	541, 570, 589, 997, 1848, 1979, 2040, 3140, 3357, 3373, 3404, 3420
\end	3444, 3463, 3482, 5218	\figurename	2357
\endcsname	437, 440, 441, 2886, 2895, 2896, 2897, 2904, 2924, 2925, 2927, 2930, 2939, 2940, 2941, 2944, 3558, 3559, 4850, 4852, 4853, 4854, 4857, 4859, 4863	figurename	12
\endgroup	438, 455, 2659, 2700, 2716, 2737, 2842, 2861, 2926, 2964, 2989, 3446, 3465, 3484, 4843, 4851, 4866	file commands:	
\endinput	438, 4848	\g_file_curr_name_str	672
\endlinechar	4822, 4823	\file_if_exist:nTF	48, 4165, 4180, 4568, 4719, 4722
\everypar	2883, 2984, 2996	\file_if_exist_input:nTF	119
\EverySelectfont	2042, 2045	\file_input:n	78, 120, 122, 546
exp commands:		\file_input_stop:	5350
\exp_after:wN	102, 149, 738, 915, 931, 933, 997, 1002, 3218, 3939, 3986, 4021, 5515	\fmtversion	5340
\exp_args:cc	3526	fmtf	33, 35, 325
\exp_args:Nc	1804, 3190, 3195, 5469	\font	4250, 4251, 4252, 4253, 4254, 4255
\exp_args:Nco	3300	fontset	7, 257, 4199
\exp_args:NNc	759, 881, 5379	\fontsize	3939
\exp_args:Nnc	109	fontspec commands:	
\exp_args:NNNo	4115	\fontspec_gset_family:Nnn	1165
\exp_args:NNo	2524	\fontspec_set_family:Nnn	1535
\exp_args:Nnx	3539	fontspec internal commands:	
\exp_args:No	384, 555, 1010, 1449, 3214, 5133, 5531	__fontspec_fontname_wrap:n	62, 1107
\exp_args:Nx	292, 583, 657, 1812, 2257, 3935, 4114	\l__fontspec_rmfamily_encoding_tl	76
\exp_last_unbraced:Ne	5480	\footnotesep	4115
\exp_last_unbraced:NNNo	152	\footnotesize	4052, 4091, 4115
\exp_last_unbraced:NNo	2033	fp commands:	
\exp_not:N	507, 585, 586, 587, 753, 754, 950, 982, 1085, 1167, 1168, 1498, 1861, 1862, 1863, 1875, 2423, 2424, 2425, 2430, 2432, 2433, 2434, 2435, 2438, 2439, 2442, 2454, 2455, 2456, 2457, 2458, 2467, 2468, 2469, 2471, 2472, 2486, 2488, 2490,	\fp_eval:n	2260, 5152
		\fp_if_nan:nTF	3899, 3926, 4112
		\fp_set:Nn	3900, 3927
		\fp_use:N	2193, 4114
		\c_nan_fp	208
		\c_zero_fp	2193

G

GBK 6, 34, 245
 \gdef . 445, 451, 3102, 3103, 3104, 3105, 3108, 3109, 3110, 3111
 \global 2764, 2983, 2986
 \globaljfont 933
 \glueexpr 3434, 3445, 3453, 3464, 3472, 3483
 group commands:
 \group_begin: 511, 723, 796, 881, 891,
 917, 943, 981, 1003, 1049, 1112, 1163, 1530, 1608, 1626,
 1643, 1855, 2002, 2082, 2429, 2437, 2502, 3026, 3126,
 3259, 3318, 4115, 5128, 5358, 5379, 5437, 5445, 5454, 5509
 \group_end: . . . 515, 524, 728, 878, 881, 896, 921, 956,
 985, 1007, 1053, 1127, 1170, 1536, 1612, 1639, 1650,
 1867, 2033, 2102, 2440, 2444, 2524, 3030, 3141, 3282,
 3322, 4115, 5139, 5358, 5379, 5457, 5504, 5532, 5566, 5590
 \group_insert_after:N 953, 954
 \gtfamily 1738, 1743

H

hbox commands:
 \hbox_set:Nn 1950, 3019, 3311
 heading 9, 298
 \heiti 4791
 hook commands:
 \hook_gput_code:nnn 5343
 \hskip 2921, 2937, 2959, 2966, 2992, 3378, 3381
 \hspace 2632, 2673, 3051, 3059, 3242
 \Huge 2697, 2734, 2839, 2858, 3620, 3651, 4069, 4108
 \huge 2655,
 2693, 2713, 2834, 3602, 3619, 3623, 3649, 3657, 4068, 4107
 hyperref 33, 349
 \hypersetup 354, 355, 1884
 \hyphenchar 4250, 4251, 4252, 4253, 4254, 4255

I

\if 444
 if commands:
 \if_bool:N 525, 577, 1801
 \if_cs_exist:N 550, 2001
 \if_false: 141
 \if_meaning:w 1978
 \if_true: 140
 \ifctexpdf 139
 \ifdim 2873, 2916, 2954, 2973
 \ifnum 442, 2628, 2650, 2669, 2692, 2773, 2832, 2899,
 2929, 2943, 3348, 3351, 3355, 3362, 3363, 3371, 3378,
 3381, 3384, 3387, 3395, 3398, 3402, 3409, 3410, 3418, 4855
 \ifodd 439, 451, 2595, 2597, 2622, 2629, 2663,
 2670, 2720, 2744, 2766, 2777, 2826, 2846, 2851, 2865,
 2877, 2888, 2904, 2917, 2955, 2974, 2978, 3438, 3457, 3476
 \ifx 437, 440, 441, 4853, 4854, 4859
 \ifzhmappdf 4851, 4880, 4904, 4928, 4985, 5017, 5041
 \ignorespaces 2999
 \immediate 4862
 indent 32, 227
 \indexname 2360
 indexname 12

\input 5101, 5165
 \insertcontinuationtext 2369, 2378, 2385
 \insertpart 3442, 3443
 \insertromanpartnumber 3432, 3589
 \insertsection 3461, 3462
 \insertsectionnumber 3451, 3687
 \insertsubsection 3480, 3481
 \insertsubsectionnumber 3470, 3716
 int commands:
 \int_case:nn 409, 4026
 \int_compare:nNnTF
 556, 741, 2234, 2277, 3775, 3791, 3807, 3897, 3924
 \int_const:Nn 1099, 1694, 1695
 \int_div_truncate:nn 84, 563
 \int_eval:n 1686, 1687, 1690, 1691, 3503
 \int_from_hex:n 555
 \int_gincr:N 1255, 1549
 \int_gset:Nn
 . . . 128, 188, 197, 198, 199, 303, 305, 373, 377, 3898, 3925
 \int_incr:N 3073
 \int_mod:nn 565
 \int_new:N 61, 127, 187, 1552
 \int_set:Nn 97, 109, 545, 554, 2231, 2272, 2529
 \int_sub:Nn 559, 2278
 \int_to_Hex:n 557, 562, 564
 \int_use:N 92, 508, 1257, 1550, 3077, 5133
 \int_zero:N 3070
 \c_zero_int 2234

int internal commands:

 __ctexpair 2960
 \interlinepenalty . 2647, 2689, 2709, 2730, 2854, 2922, 2960

iow commands:

 \iow_close:N 5122
 \iow_indent:n 5313
 \iow_new:N 5124
 \iow_now:Nn 5134
 \iow_open:Nn 5119
 \itemsep 4049, 4060, 4088, 4099
 \itshape 1738, 1739, 1743, 1744

J

\jfam 1041, 1100, 1735

K

\kaishu 4791
 \kanjifamily 1729, 1843, 1844, 1845
 \kanjifamilydefault 1846
 kernel internal commands:
 \l__kernel_expl_bool 86
 keys commands:
 \l_keys_choice_tl 293
 \keys_define:nn 66, 69, 1104, 1436, 1546, 1700
 \keys_if_exist:nnTF 1698
 \l_keys_key_tl 166, 168, 182
 \keys_set:nn 71, 73
 \keys_set_known:nn 3855
 \keys_set_known:nnN 1116

L

`\labelformat` 3539, 3563
`\languagealias` 3884, 3889, 3910, 3915
`\LARGE` 4067, 4106
`\Large` 2651, 3600, 3606, 3674, 3678, 4066, 4105
`\large` 3699, 4065, 4104
`\lastbox` 2987
`\leavevmode` 2590, 2869
left commands:
`\c_left_brace_str` 5501
`\leftmargin` 4046, 4057, 4085, 4096
`\leftmargini` 4046, 4057, 4085, 4096
`\let` 447, 450, 2902, 2911, 4039, 4078, 4846, 4852
`\linespread` 4114
`linespread` 11, 207
`linestretch` 11, 2198
`\linewidth` 2232, 2240, 2271
`\lishu` 4791
`\listfigurename` 2355
`listfigurename` 12
`\listtablename` 2356
`listtablename` 12
`\LoadClass` 417, 422, 427, 432
`\long` 4860
`\ltjdefcharrange` . 681, 682, 683, 685, 690, 691, 695, 698, 699
`LTJFONTUID` 1546
`\ltjgetparameter` 1955, 1982, 2060, 2074
`\ltjsetkanjiskip` 1722
`\ltjsetmathletter` 1051
`\ltjsetparameter` 701, 704, 710, 1719
`\ltjsetxkanjiskip` 1724
lua commands:
`\lua_now:n` 798, 2084
`\LuaTeXjversion` 865

M

`\makeatletter` 766, 770
`\MAKESPA` 5125, 5166
`\markboth` 2563, 2657, 2686
`\mathgroup` 1041
max commands:
`\c_max_char_int` 1695
`\mcfamily` 1743
`\mddefault` 1086, 1090, 1096, 1424, 1429
mode commands:
`\mode_if_math:TF` 1035
msg commands:
`\msg_critical:nnn` 53, 672, 4139
`\msg_error:nn` 4219
`\msg_error:nnn` 19, 22, 38, 41, 663, 676, 679, 2342, 3940, 4022
`\msg_error:nnnn` 4185, 4214
`\msg_fatal:nn` 5107
`\msg_new:nnn` . 165, 167, 181, 183, 399, 666, 1157, 1223,
1305, 2373, 3829, 4140, 4193, 4228, 5105, 5309, 5329, 5421
`\msg_new:nnnn` 12, 30, 45, 133, 2344, 3942, 4195, 4222
`\msg_warning:nn` 250, 404, 1300, 2123, 2134, 2395

`\msg_warning:nnn` 170, 1218, 3848, 4206, 5315, 5328, 5420
`\msg_warning:nnnn` 178, 1153, 4175

N

`\newCJKfontfamily` 1324
`\NewDocumentCommand`
..... 1184, 1319, 1324, 1330, 1340, 1346, 1352,
1359, 1365, 1371, 1377, 1574, 1674, 1728, 2141, 2144,
2154, 2181, 2256, 2293, 2300, 2321, 2323, 2492, 3934,
4119, 4792, 4793, 4794, 4795, 4797, 4798, 4800, 4810, 4811
`\newfontfeature` 1101, 1102, 1103
`\newpage` 2746, 2751
`\ngostype` 44
`\nobreak` 2651,
2660, 2717, 2834, 2839, 2843, 2858, 2862, 2975, 3601, 3650
`\nobreakspace` 2651, 2693
`nocap` 33, 321
`\noexpand` 4844, 4846, 4847, 4848
`nofonts` 32
`noindent` 32
`nopunct` 33
`\normalem` 1924
`\normalfont` 1356, 1363, 1369, 1407,
1827, 1849, 2527, 2647, 2689, 2709, 2730, 2829, 2854, 3081
`\normalsize` 3733, 3750, 3764, 4033, 4072, 4110, 4118
`nospace` 33
`nozhmap` 32
`\null` 2620, 2749
`\numberline` 2783, 2930, 2944, 3059, 3067, 3253, 3318

O

`\onecolumn` 2615

P

`\pagestyle` 3875
`\par` 2554, 2591, 2651, 2657, 2658, 2693,
2697, 2699, 2713, 2715, 2734, 2736, 2834, 2839, 2841,
2858, 2860, 2870, 2925, 2963, 2975, 3433, 3434, 3442,
3445, 3452, 3453, 3461, 3464, 3471, 3472, 3480, 3483,
3579, 3601, 3612, 3630, 3638, 3650, 3690, 3692, 3724, 3726
`paragraph/afterskip` 34
`paragraph/beforeskip` 34
`paragraph/numbering` 16
`paragraph/runin` 22
`\parindent` 2252, 2253,
2275, 2296, 2298, 2303, 2304, 2647, 2709, 2829, 2854, 3809
`\parsep` 4048, 4049, 4059, 4060, 4087, 4088, 4098, 4099
`\part` 2588
`part/beforeskip` 34
`part/fixbeforeskip` 34
`part/numbering` 16
`part/pagestyle` 21, 2502
`\partmark` 2562, 2645, 2687
`\partname` 2651, 2693, 3432, 3584
`\PassOptionsToClass` 416, 421, 426, 431
`\PassOptionsToPackage` 526, 1888, 2306, 5317

`\pdfmapline` 4929, 4930, 4931, 4932, 4933, 4934,
 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943,
 4944, 4945, 4946, 4947, 4948, 4949, 4950, 4951, 4952,
 4953, 4954, 5042, 5043, 5044, 5045, 5046, 5047, 5048,
 5049, 5050, 5051, 5052, 5053, 5054, 5055, 5056, 5057,
 5058, 5059, 5060, 5061, 5062, 5063, 5064, 5065, 5066, 5067
`\pdfoutput` 442, 4855
`\pdfstringdefDisableCommands` 4122
`\pingfang` 4791
prg commands:
`\prg_do_nothing:`
 471, 1045, 1524, 1609, 2000, 2110, 3043, 5542
`\prg_generate_conditional_variant:Nnn` 1210
`\prg_new_conditional:Npnn` 959, 1975
`\prg_new_protected_conditional:Npnn` 1197, 5512
`\prg_return_false:` ... 962, 1207, 1979, 1983, 1987, 5522
`\prg_return_true:` 961, 1200, 1205, 1979, 1983, 1987, 5527
`\primitive` 5218
`\ProcessKeysOptions` 396
`\proofname` 2347, 2362
`proofname` 13
prop commands:
`\prop_clear:N` 1114
`\prop_const_from_keyval:Nn` 3506
`\prop_get:NnN` 1292
`\prop_get:NnNTF`
 ... 1199, 1251, 1289, 1517, 1563, 3501, 3938, 3993, 4020
`\prop_gpop:NnNTF` 1142, 1146, 1644
`\prop_gput:Nnn` . 1120, 1121, 1167, 1279, 1632, 1668, 3955
`\prop_gremove:Nn` 1151
`\prop_if_empty:NTF` 1213, 1388, 1468
`\prop_map_break:n` 1402
`\prop_map_inline:Nn` 1400, 1476
`\prop_new:N` 1137, 1138, 1139, 1284, 1458, 1652, 1673, 3949
`\prop_put:Nnn` 1455
`\prop_remove:Nn` 1454
`\protect` 2576, 2582, 2783,
 2804, 2805, 2930, 2944, 3059, 3067, 3252, 3264, 3266,
 3270, 3272, 3276, 3278, 3286, 3291, 3303, 3304, 5473, 5474
`\ProvideDocumentCommand` 2562
`\ProvidesFile` 4858, 4867, 4869, 4871, 4873, 4875, 4877
`\providetranslation`
 5223, 5224, 5225, 5226, 5227, 5228, 5229,
 5230, 5231, 5232, 5233, 5234, 5235, 5236, 5237, 5238,
 5239, 5240, 5241, 5242, 5243, 5244, 5245, 5246, 5247,
 5248, 5249, 5250, 5251, 5252, 5253, 5254, 5255, 5256,
 5257, 5258, 5259, 5260, 5261, 5262, 5263, 5264, 5265,
 5266, 5267, 5268, 5269, 5270, 5271, 5272, 5273, 5274,
 5275, 5276, 5277, 5278, 5279, 5280, 5281, 5282, 5283,
 5284, 5285, 5286, 5287, 5288, 5289, 5290, 5291, 5292,
 5293, 5294, 5295, 5296, 5297, 5298, 5299, 5300, 5301, 5302
`punct` 10, 33, 284, 2126
`\punctstyle` 574, 2131

Q

`\quad` . 2895, 3348, 3349, 3351, 3352, 3355, 3356, 3368, 3372,

3379, 3382, 3385, 3388, 3395, 3396, 3398, 3399, 3402,
 3403, 3415, 3419, 3607, 3658, 3667, 3700, 3734, 3751, 3765

quark commands:

`\q_mark` 1477, 1485, 5516, 5519
`\q_stop` 150,
 153, 996, 999, 1002, 1005, 5481, 5486, 5495, 5516, 5519
`\quotation` 4132

R

`\raggedright` 2647, 2709, 2829, 2854, 3599, 3648
`\refname` 2363, 2368, 2375, 2384, 2392, 2393
`refname` 13
`\refstepcounter` ... 2631, 2672, 2779, 2906, 3521, 3566, 3567
`\relax` 437, 440, 441, 444, 2669, 2692, 2871, 2907, 2915, 2921,
 2937, 2953, 2966, 2972, 2991, 3378, 3381, 3434, 3445,
 3453, 3464, 3472, 3483, 4817, 4822, 4825, 4853, 4854, 4859
`\renewcommand` 2589, 2605, 2759
`\RenewDocumentCommand` 3567
`\RequirePackage`
 27, 56, 57, 59, 330, 336, 347, 357, 527, 530, 531,
 532, 656, 674, 677, 790, 795, 1798, 1897, 1906, 2307, 3929
`\RequirePackageWithOptions` 5318
`resetalternatfont` 30, 1615

reverse commands:

`\reverse_if:N` 577, 1801
`\rmdefault` 1861
`\rmfamily` 1824

S

scan commands:

`\scan_stop:`
 ... 71, 150, 754, 933, 1041, 1610, 2223, 2232, 5129, 5360
`scheme` 10, 309
`\scriptsize` 4063, 4102
`\secdef` 2602, 2624, 2771
`secnumdepth` 16, 3490
`section/afterskip` 34
`section/beforeskip` 34
`section/numbering` 16
`section/runin` 22
`\sectionname` 3451, 3684
`\selectfont` 1191, 1635,
 1648, 1729, 1824, 1825, 1826, 2032, 2039, 2053, 2183, 3939

seq commands:

`\seq_const_from_clist:Nn` . 478, 2404, 2414, 3116, 5109
`\seq_gpop:NN` 101
`\seq_gpop:NNTF` 909
`\seq_gpush:Nn` 83, 904
`\seq_gput_left:Nn` 2410, 2411
`\seq_gput_right:Nn` 1217, 3961
`\seq_gset_eq:NN` 2409
`\seq_if_in:NnTF` 484, 1215
`\seq_map_function:NN` 3185, 3307
`\seq_map_inline:Nn`
 ... 2519, 3071, 3119, 3129, 3138, 3161, 3244, 3532, 5132
`\seq_new:N` 112, 912, 1222, 2408, 3950
`\seq_use:Nnnn` 3947

`\setbox` 2987
`\setCJKfamilyfont` 1236, 1324, 4379,
 4381, 4383, 4385, 4421, 4422, 4423, 4424, 4443, 4445,
 4447, 4449, 4492, 4494, 4496, 4498, 4511, 4512, 4513,
 4514, 4515, 4516, 4557, 4559, 4560, 4561, 4562, 4563,
 4591, 4592, 4593, 4594, 4595, 4596, 4637, 4638, 4639,
 4640, 4651, 4652, 4653, 4654, 4655, 4656, 4657, 4673,
 4675, 4677, 4713, 4714, 4715, 4736, 4737, 4738, 4739,
 4740, 4742, 4743, 4782, 4783, 4784, 4785, 4786, 4787, 4788
`\setCJKmainfont` 1232, 1352, 4369, 4417, 4433,
 4482, 4507, 4553, 4583, 4633, 4643, 4667, 4710, 4731, 4779
`\setCJKmathfont` 1352
`\setCJKmonofont` 1234, 1352, 4377, 4420, 4441,
 4490, 4510, 4556, 4590, 4636, 4650, 4671, 4712, 4735, 4781
`\setCJKromanfont` 1358, 1384
`\setCJKsansfont` 1233, 1352, 4375, 4419, 4439,
 4488, 4509, 4555, 4589, 4635, 4649, 4669, 4711, 4733, 4780
`\setcounter` 3502, 3503
`\SetSymbolFont` 1097, 1428, 1432, 1734
`\sfdefault` 1862
`\sffamily` 1825
`\shapedefault`
 ... 1086, 1090, 1092, 1096, 1098, 1424, 1426, 1429, 1433
skip commands:
`\skip_horizontal:N` 1965, 3029
`\skip_if_eq:nnTF` 1982, 1986, 2059, 2573, 2579
`\skip_new:N` 1974, 2072, 2558
`\skip_set:Nn` 1955, 1958,
 2063, 2073, 2224, 2236, 2280, 2288, 2560, 2572, 2578, 3027
`\skip_set_eq:NN` 1972, 2065
`\skip_sub:Nn` 2549, 2556
`\skip_use:N` 2576, 2582
`\skip_zero:N` 2245
`\c_zero_skip` 2460, 2461, 2510, 2511, 2573, 2579
`\small` 4041, 4080
`\songti` 4791
`\space` 442, 2780, 2834, 3584, 3646, 3684, 3715, 3816
`space` 10, 33, 289, 2105
`\special` 1765, 1766, 1769, 1770, 1778, 1779,
 1782, 1783, 1904, 1918, 4883, 4884, 4885, 4886, 4887,
 4888, 4889, 4890, 4891, 4892, 4893, 4894, 4895, 4896,
 4897, 4898, 4899, 4900, 4907, 4908, 4909, 4910, 4911,
 4912, 4913, 4914, 4915, 4916, 4917, 4918, 4919, 4920,
 4921, 4922, 4923, 4924, 4956, 4957, 4958, 4959, 4960,
 4961, 4962, 4963, 4964, 4965, 4966, 4967, 4968, 4969,
 4970, 4971, 4972, 4973, 4974, 4975, 4976, 4977, 4978,
 4979, 4980, 4981, 4988, 4989, 4990, 4991, 4992, 4993,
 4994, 4995, 4996, 4997, 4998, 4999, 5000, 5001, 5002,
 5003, 5004, 5005, 5006, 5007, 5008, 5009, 5010, 5011,
 5012, 5013, 5020, 5021, 5022, 5023, 5024, 5025, 5026,
 5027, 5028, 5029, 5030, 5031, 5032, 5033, 5034, 5035,
 5036, 5037, 5069, 5070, 5071, 5072, 5073, 5074, 5075,
 5076, 5077, 5078, 5079, 5080, 5081, 5082, 5083, 5084,
 5085, 5086, 5087, 5088, 5089, 5090, 5091, 5092, 5093, 5094
`\SplitArgument` 1575, 1675, 2493

str commands:

`\c_backslash_str` 5500
`\c_colon_str` 1028, 1029, 1030
`\c_dollar_str` 4719
`\str_case:nnTF` 1859, 5471, 5486
`\str_case_e:nnTF` 1230, 1241
`\str_const:Nn` 43, 50
`\str_if_empty:NTF` 5557, 5593, 5602
`\str_if_eq:nnTF` 529, 573, 578, 1074, 1392,
 1893, 1901, 3880, 3907, 3921, 4205, 4208, 4225, 4230, 5477
`\str_if_eq_p:nn` 4172, 4173
`\str_lowercase:n` 503
`\str_new:N` 5506, 5507, 5508
`\str_set:Nn` 5524, 5525, 5526
`\strutbox` 4116
`sub3section` 9, 301
`sub4section` 9, 301
`subparagraph/afterskip` 34
`subparagraph/beforeskip` 34
`subparagraph/numbering` 16
`subparagraph/runin` 22
`subsection/afterskip` 34
`subsection/beforeskip` 34
`subsection/numbering` 16
`subsection/runin` 22
`\subsectionname` 3470, 3715
`subsubsection/afterskip` 34
`subsubsection/beforeskip` 34
`subsubsection/numbering` 16
`subsubsection/runin` 22
`\symmincho` 1735
sys commands:
`\c_sys_engine_str` 44
`\sys_if_engine_pdftex:TF` ... 247, 333, 403, 4144, 4150
`\sys_if_engine_pdftex_p:` 4803
`\sys_if_engine_uptex:TF` 4146, 4152
`\sys_if_engine_uptex_p:` 4804
`\sys_if_engine_xetex:TF` 327, 5103
`\sys_if_output_pdf:TF` 139, 513, 1896, 4151
`\sys_if_platform_windows:TF` 4156

T

`\tablename` 2358
`tablename` 12
`\TeX` 4120
TeX and L^AT_EX 2_ε commands:
`\@ifdefinable` 731
`\@italiccorr` 1720
`\@par` 2922, 3668, 3701, 3735, 3779, 3795
`\@afterheading` 2665, 2722, 2811, 2821, 2980
`\@afterindentfalse` 2600, 2765, 2769, 2874, 2880
`\@afterindenttrue` 2598, 2767, 2872, 2878
`\@Alph` 3812, 3817
`\@break@tfor` 450
`\@chapapp` 2780, 2834, 3364, 3384, 3411
`\@chapter` 2771, 2772

\@chinese	2309, 2318	\@sect	2893, 2898
\@classoptionslist	47, 445, 451, 453	\@setfontsize	4025
\@clubpenalty	2995	\@sffamilyhook	1825
\@currenthref	3150	\@spart	2602, 2624, 2703
\@currentlabel	3523, 3527	\@sssect	2892, 2950
\@currentxt	4, 142, 5308, 5387, 5393	\@startsection	99, 102, 110, 2868, 3076
\@currname	4, 142, 5308, 5388, 5394	\@svsec	2902, 2907, 2911, 2921, 2923, 2937, 2938
\@dblarg	2893	\@svsechd	2935, 2966, 2967, 2989
\@defaultfamilyhook	1827	\@tempdima	3312, 3314
\@empty	109, 2902, 2911	\@tempskipa	2871, 2873, 2874, 2885, 2915, 2916, 2953, 2954, 2972, 2973, 2976, 2991, 2992
\@endpart	2701, 2738, 2740	\@tempswafalse	2618
\@EverySelectfont@Init	2001, 2015, 2027	\@tempswattrue	2616
\@firstofone	56	\@text@composite@x	54, 781
\@firstoftwo	3303	\@tfor	448
\@gobble	56	\@topnewpage	2808, 2818
\@hangfrom	2921, 2959	\@topnum	2764
\@ifclassloaded	3825	\@ttfamilyhook	1826
\@ifdefinable	731	\@xssect	2949, 2970, 2971
\@ifl@t@r	5340	\@zhdig	2317
\@ifpackagelater	37, 40, 662, 675, 678, 703, 1931	\@zhnum	2309, 2316
\@ifpackageloaded	18, 21, 671, 785, 1881, 3224, 3329, 3543, 5323, 5336	\abstractname	12
\@ifpackagewith	3169, 3178, 3229, 3390	\add@unicode@accent	54, 782
\@ifstar	2891	\addCJKfontfeature	71
\@listI	4039, 4078	\AfterEndPreamble	141
\@listi	4039, 4046, 4057, 4078, 4085, 4096	\algorithmname	13
\@M	2647, 2689, 2709, 2730, 2854, 2922, 2960, 2988	\appendixname	12, 25
\@makechapterhead	2808, 2810, 2823	\AtBeginDocument	40, 49
\@makeschapterhead	2818, 2820, 2848	\AtBeginDvi	47
\@minus	3669, 3702, 3736, 3752, 3766, 4035, 4037, 4043, 4045, 4047, 4048, 4054, 4056, 4058, 4059, 4074, 4076, 4082, 4084, 4086, 4087, 4093, 4095, 4097, 4098	\AtBeginShipoutFirst	47
\@namedef	3264, 3266, 3270, 3272, 3276, 3278, 3286, 3291	\AtEndOfClass	41
\@nameuse	3264, 3266, 3276, 3278, 3286	\AtEndOfPackage	41
\@ne	447, 3222, 3351, 3381, 3398	\AtEndOfPackageFile*	141
\@nil	982	\AtEndPreamble	141
\@nobreakfalse	2982	\baselineskip	11, 11, 81, 85
\@noskipsecfalse	2986	\bfseries	51
\@noskipsectrue	2983	\bibname	12
\@onlypreamble	40, 475, 540, 1380, 1381, 1382, 1383, 1384, 1385, 1794, 1795, 1796, 1797, 4198	\c@chapter	3817
\@part	2602, 2624, 2626	\c@secnumdepth	2628, 2650, 2669, 2692, 2773, 2832, 2899, 2929, 2943, 3348, 3351, 3355, 3362, 3363, 3371, 3378, 3381, 3384, 3387, 3395, 3398, 3402, 3409, 3410, 3418
\@pkgextension	6, 7, 8, 9, 20, 23, 673, 5387, 5393	\c@section	3812
\@plus	3613, 3614, 3669, 3670, 3702, 3703, 3736, 3737, 3752, 3766, 3780, 3796, 4035, 4036, 4037, 4043, 4044, 4045, 4047, 4048, 4054, 4055, 4056, 4058, 4059, 4074, 4075, 4076, 4082, 4083, 4084, 4086, 4087, 4093, 4094, 4095, 4097, 4098	\caption	34
\@popfilename	142, 5390	\catcode	49, 132, 142
\@pushfilename	142	\ccwd	11, 11, 27, 27, 33, 33, 40, 79, 80, 80, 84, 84, 85, 85
\@reset@options	142, 5384	\chapter	9, 17, 21, 26, 26, 26, 91, 98, 114
\@rmfamilyhook	76, 1824	\chaptermark	13
\@schapter	2771, 2813	\chaptername	17
\@secCNTformat	2894, 2907	\char	55
\@secondoftwo	3304	\chardef@text@cmd	54, 779
\@secpenalty	2885, 3672, 3705, 3739, 3753, 3767	\CheckCommand	80
		\Chinese	33
		\chinese	28, 33, 87
		\CJK@@ignorespaces	50, 590
		\CJK@encoding	62, 63, 70, 888, 892, 976, 1081, 1135, 1423, 1426, 1428, 1432, 1572, 1582, 1583, 1601, 1602

\CJK@envStart	50, 50, 585	\CTEX@fixtopskip	2544, 2622, 2826, 2851
\CJK@family	57,	\CTEX@getttitle
879, 893, 944, 976, 1188, 1634, 1644, 1646, 2053, 2054		2643, 2684, 2707, 2728, 2801, 2816, 2914, 2952, 3156	
\CJK@ignorespaces	50, 51, 594, 598	\CTEX@hang	3007, 3039, 3046
\CJK@input	50, 542	\CTEX@heading@format@initial	2525, 2648,
\CJK@loadBinding	49, 50, 581	2690, 2710, 2731, 2830, 2855, 2919, 2957, 3435, 3454, 3473	
\CJK@loadEncoding	49	\CTEX@heading@glue	2993, 3024
\CJK@makeActive	50, 580	\CTEX@headinghang 2653, 2695, 2712, 2733, 2837, 2857, 3000	
\CJK@plane	49, 81, 492, 493, 503, 2042	\CTEX@headingskip	2561, 2596,
\CJK@surr	49, 550	2623, 2664, 2721, 2745, 2827, 2847, 2852, 2866, 2889, 2979	
\CJK@upperReset	50	\CTEX@hyperheadinghook	3124, 3154
\CJKfamily	63, 74	\CTEX@ifnamefalse	2564, 2634, 2639, 2675,
\CJKfamilydefault	50, 50, 67, 77, 77, 77, 77	2680, 2705, 2726, 2785, 2791, 2797, 2814, 2900, 2909, 2951	
\CJKglue	84, 85, 119	\CTEX@ifnametrue	2564, 2630, 2671, 2778, 2905
\CJKhook	51, 51	\CTEX@indentbox	2542, 2938, 2967, 3437, 3456, 3475
\CJKrmdefault	49	\CTEX@makeanchor	2635,
\CJKsymbol	81	2640, 2676, 2681, 2786, 2910, 3114, 3122, 3134, 3139, 3145	
\CJKunderdotbasesep	33	\CTEX@makeanchor@chapter	2792, 2798
\CJKunderline	33	\CTEX@makeanchor@schapter	2815
\clearpage	50	\CTEX@makeanchor@sect	2901
\contentsname	12, 12	\CTEX@makeanchor@spart	2706, 2727
\CS	58	\CTEX@makeanchor@ssect	2892
\CTEX@add	447, 450, 451	\CTEX@part@afterindent	2597
\CTEX@add@unicode@accent	750, 782	\CTEX@part@aftername	2654, 2696, 3439
\CTEX@addloflotskip	2569, 2806	\CTEX@part@afterskip	2662, 2719, 2743, 3445
\CTEX@addtocline ...	2586, 2644, 2685, 2802, 2933, 2947	\CTEX@part@aftertitle ...	2658, 2699, 2715, 2736, 3443
\CTEX@afterindent	2877, 3037, 3044	\CTEX@part@beforeskip	2594, 2621, 3434
\CTEX@aftertitle	2963, 2968, 3036, 3043	\CTEX@part@break	2592, 2611
\CTEX@alchar	719, 742	\CTEX@part@fixskip	2595, 2622, 2663, 2720, 2744
\CTEX@appendix@number	3091, 3103, 3109	\CTEX@part@format	2649, 2691, 2711, 2732, 3436
\CTEX@appendix@numbering	3092, 3105, 3111	\CTEX@part@indent	3437
\CTEX@beginallalchar	719, 746, 752, 767, 791	\CTEX@part@numbering	2629, 2670, 3438
\CTEX@chapter@afterindent	2766	\CTEX@part@pagestyle	2506, 2613
\CTEX@chapter@aftername	2838	\CTEX@part@titleformat ..	2656, 2698, 2714, 2735, 3443
\CTEX@chapter@afterskip	2845, 2864	\CTEX@part@tocline	3048
\CTEX@chapter@aftertitle	2841, 2860	\CTEX@partname	2654, 2696, 3439
\CTEX@chapter@beforeskip	2825, 2850	\CTEX@patch@text@cmd	757, 783
\CTEX@chapter@break	2762	\CTEX@patch@tunec	777
\CTEX@chapter@fixskip	2826, 2846, 2851, 2865	\CTEX@popfilename@hook	5390
\CTEX@chapter@format	2831, 2856	\CTEX@postappendix	3096, 3104, 3110
\CTEX@chapter@lofskip	2508	\CTEX@postchapter	3110
\CTEX@chapter@lotskip	2509	\CTEX@postsection	3104
\CTEX@chapter@numbering	2777, 3111	\CTEX@preappendix	3095, 3102, 3108
\CTEX@chapter@pagestyle	2507, 2763	\CTEX@prechapter	3108
\CTEX@chapter@titleformat	2840, 2859	\CTEX@presection	3102
\CTEX@chapter@tocline	3048	\CTEX@reset@options@hook	5384
\CTEX@chaptername	2838	\CTEX@rmfamilyhook	76, 1802, 1834, 1843
\CTEX@chardef@text@cmd	729, 779	\CTEX@runin	2917, 2955, 2974, 3040, 3047
\CTEX@defaultfamilyhook	1837	\CTEX@save@appendix	3097, 3100
\CTEX@document@left@hook	5357, 5363	\CTEX@save@refstepcounter	3566, 3569
\CTEX@document@right@hook	5359, 5365	\CTEX@section@aftername	3458
\CTEX@endallalchar	719, 748, 755, 771, 792	\CTEX@section@afterskip	3464
\CTEX@Family@CMap	489, 517, 520	\CTEX@section@aftertitle	3462
\CTEX@fixheadingskip		\CTEX@section@beforeskip	3453
2547, 2552, 2595, 2663, 2720, 2744, 2846, 2865, 2888, 2978		\CTEX@section@format	3455
\CTEX@fixskip	2888, 2978, 3038, 3045	\CTEX@section@indent	3456

\CTEX@section@numbering	3105, 3457	\em	74
\CTEX@section@titleformat	3462	\endlinechar	39, 49
\CTEX@sectionhang	2923, 2961, 3000	\escapechar	60
\CTEX@sectionname	3458	\EverySelectfont	80
\CTEX@selectfont@hook	1995, 2042, 2045	\external@font	930, 931, 933
\CTEX@selectfont@save	2006, 2021, 2025, 2028	\extract@font	58, 918
\CTEX@setcurrentlabel@n	3521, 3570	\f@baselineskip	2223
\CTEX@setheadingskip	2559, 2594, 2621, 2662, 2719, 2743, 2825, 2845, 2850, 2864, 2887, 2977	\f@encoding	892, 936, 1038
\CTEX@ssfamlyhook	1835, 1844	\f@family	57, 893, 936, 944
\CTEX@subsection@aftername	3477	\f@nch@initialise	3375
\CTEX@subsection@afterskip	3483	\f@series	888, 976
\CTEX@subsection@aftertitle	3481	\f@shape	888, 976
\CTEX@subsection@beforeskip	3472	\f@size	888, 894, 971, 979, 982, 983, 1951, 1992, 3315
\CTEX@subsection@format	3474	\fangsong	8
\CTEX@subsection@indent	3475	\fbox	18
\CTEX@subsection@numbering	3476	\figurename	12
\CTEX@subsection@titleformat	3481	\font	56, 59
\CTEX@subsectionname	3477	\font@name	57, 58, 897, 904, 905, 910, 915, 933, 934, 951, 1019, 1020, 1021, 1022, 1023, 1030
\CTEX@text@composite@x	744, 781	\fontcharwd	80
\CTEX@textcmd	764, 765, 769, 772	\footnotesep	119, 119
\CTEX@thechapter	3109	\footnotesize	119
\CTEX@thesection	3103	\get@external@font	927
\CTEX@titleformat@n	99, 2962, 2968, 3035, 3042	\getanddefine@fonts	1008
\CTEX@titlepslabel@clear	3273, 3283	\globaldefs	57
\CTEX@titlepslabel@set	3267, 3279, 3283	\globaljfont	58
\CTEX@toc@width@n	3309, 3324, 3336	\H@old@chapter	3128
\CTEX@todayold	2325, 2330	\H@refstepcounter	3544
\CTEX@ttfamilyhook	1836, 1845	\hbox	22, 91
\CTEX@update@sectionformat@n	2876, 3033	\heiti	8, 131
\CTEX@verbatim@font@hook	1714, 1718	\hrule	91
\ctex@zhmap@endinput	5097	\Hy@chapapp	2786, 2792, 2798, 2815
\CTEXdigits	28	\Hy@driver	1901
\CTEXifname	25, 26	\Hy@MakeCurrentHrefAuto	3147
\CTEXnumber	28, 28	\Hy@org@chapter	3128
\CTEXnumberline	25	\Hy@raisedlink	3148
\CTEXoptions	33, 34	\hyper@anchorend	3151
\ctexset	5, 5, 6, 8, 12, 15, 30, 30, 33, 33, 121	\hyper@anchorstart	3150
\CTEXsetup	33	\if@mainmatter	2775, 2833, 3362, 3409, 3843, 3844
\CTEXthechapter	13, 14, 26, 103	\if@nobreak	2882
\CTEXthesection	13, 17, 106	\if@noskipsec	2590, 2869, 2985
\CTEXunderdotbasesep	33	\if@openright	2606, 2748, 2760, 3616, 3644
\CTEXunderline	33	\if@tempswa	2754
\curr@fontshape	894, 928, 937, 941, 983	\if@twocolumn	2614, 2807, 2817
\DeclareAlternateKanjiFont	70	\if@twoside	2747, 3346, 3369, 3393, 3416
\DeclareFontEncoding	60	\iffalse	60
\DeclareFontFamily	48, 58	\ifHy@implicit	3127
\DeclareFontShape	58	\ifin@	60, 450, 997
\DeclareFontShape@	70, 1533	\iftrue	60
\DeclareRobustCommand	143	\in@	449
\DeclareTextCommand	54	\indexname	12
\DeclareTextSymbol	54	\insertcontinuationtext	13
\DeclareUnicodeAccent	54	\itdefault	59
\DeclareUnicodeComposite	54	\itshape	52
\define@newfont	58, 920	\jfam	74
\do@subst@correction	58, 59, 919	\kaishu	8

<code>\kanjifamilydefault</code>	76	<code>\PassOptionsToClass</code>	46
<code>\kanjiskip</code>	81	<code>\PassOptionsToPackage</code>	78
<code>\labelformat</code>	109, 109, 109	<code>\pdfmapline</code>	124
<code>\leftmark</code>	13	<code>\pickup@font</code>	57
<code>\linewidth</code>	84	<code>\pickup@jfont</code>	924
<code>\lishu</code>	8	<code>\pingfang</code>	8, 131
<code>\listfigurename</code>	12	<code>\ProcessKeysOptions</code>	31, 41
<code>\listtablename</code>	12	<code>\proofname</code>	13, 87
<code>\ltj@@does@alt@set</code>	59	<code>\protected</code>	104
<code>\ltj@@getjfontnumber</code>	60, 991	<code>\protected@edef</code>	2322, 2324, 2907, 3523
<code>\ltj@@set@stackfont</code>	1028, 1029, 1030	<code>\ProvidesFile</code>	132
<code>\ltj@callalchar</code>	725	<code>\ps@headings</code>	106, 3345, 3394, 3397, 3401, 3407, 3417
<code>\ltj@curjfnt</code>	58, 81	<code>\refname</code>	12, 13
<code>\ltj@pickup@altfont@auxy</code>	56, 977	<code>\refstepcounter</code>	109
<code>\ltj@pickup@altfont@copy</code>	56, 56, 989	<code>\refstepcounter@noarg</code>	3546
<code>\ltj@setpar@global</code>	74, 1027, 1722, 1724	<code>\refstepcounter@optarg</code>	3547
<code>\ltj@tempcntc</code>	60, 992	<code>\relax</code>	49, 84
<code>\ltjalchar</code>	55	<code>\rightmark</code>	13
<code>\ltjsetkanjiskip</code>	74	<code>\rmfamily</code>	52, 76, 76, 76
<code>\ltjsetparameter</code>	74	<code>\roman</code>	28
<code>\ltjsetxkanjiskip</code>	74	<code>\scantokens</code>	143
<code>\m@ne</code>	2628, 2650, 2773, 2832, 3355, 3362, 3363, 3384, 3409, 3410, 4250, 4251, 4252, 4253, 4254, 4255	<code>\scr@new@selectfont</code>	2009, 2024
<code>\markboth</code>	13	<code>\scr@selectfont</code>	2021, 2025
<code>\markright</code>	13	<code>\scriptfont@name</code>	1022, 1029
<code>\math@bgroup</code>	1037	<code>\section</code>	9, 15, 16, 17, 17, 22, 22, 23, 23, 24, 24, 26, 26, 34, 88, 98, 106
<code>\math@egroup</code>	1042	<code>\section*</code>	16
<code>\math@fonts</code>	1025	<code>\sectionmark</code>	13
<code>\mathbb</code>	120	<code>\sectionname</code>	17
<code>\maxdimen</code>	11, 84	<code>\sectionpage</code>	15, 107
<code>\meaning</code>	143, 144, 144, 145	<code>\selectfont</code>	59, 75, 80, 80, 81, 84
<code>\newcommand</code>	143	<code>\setCJKfamilyfont</code>	29, 29
<code>\newrobustcmd</code>	143	<code>\setCJKmainfont</code>	30
<code>\newtitlemark</code>	104	<code>\setcounter</code>	108
<code>\nfss@catcodes</code>	62	<code>\sf@size</code>	1021
<code>\noindent</code>	90	<code>\sffamily</code>	52
<code>\normalfont</code>	76	<code>\size@update</code>	84, 85, 2036, 2037, 2183, 2197
<code>\normalsize</code>	8, 115	<code>\sldefault</code>	59
<code>\nouppercase</code>	49	<code>\songti</code>	8
<code>\NR@getttitle</code>	3160	<code>\special</code>	6, 47, 78
<code>\numberline</code>	105	<code>\split@name</code>	982
<code>\p@</code>	1951, 1992, 2693, 2804, 2805, 2824, 2834, 2844, 2849, 2863, 3315, 3612, 3640, 3641, 3642, 3643, 3650, 4035, 4036, 4037, 4043, 4044, 4045, 4047, 4048, 4054, 4055, 4056, 4058, 4059, 4074, 4075, 4076, 4082, 4083, 4084, 4086, 4087, 4093, 4094, 4095, 4097, 4098	<code>\SplitArgument</code>	90
<code>\pagenumbering</code>	28	<code>\ssf@size</code>	1023
<code>\pagestyle</code>	10	<code>\strutbox</code>	119
<code>\par@update</code>	80, 2012, 2016, 2020	<code>\subparagraph</code>	9, 22, 40
<code>\paragraph</code>	9, 22, 40	<code>\subsectionname</code>	17
<code>\parindent</code>	11, 33, 84, 86, 91, 99, 119	<code>\subsectionpage</code>	15, 108
<code>\parskip</code>	91	<code>\subst@correction</code>	947, 954
<code>\part</code>	15, 21, 23, 24, 91, 98	<code>\tablename</code>	12
<code>\partmark</code>	91	<code>\textfont@name</code>	1020, 1028
<code>\partname</code>	17	<code>\tf@size</code>	1019
<code>\partpage</code>	15, 107	<code>\thechapter</code>	13, 26, 26
		<code>\thesection</code>	13, 17, 26, 106
		<code>\titleformat</code>	102
		<code>\titlespacing</code>	102
		<code>\today</code>	11, 11

<code>\trans@languagepath</code>	3892, 3918	<code>\tex_parindent:D</code>	2534, 2539, 3010
<code>\ttfamily</code>	52	<code>\tex_parskip:D</code>	2556
<code>\ttl@a</code>	3242, 3250	<code>\tex_pdffontattr:D</code>	507
<code>\ttl@chapterout</code>	3237	<code>\tex_pdflastobj:D</code>	508
<code>\ttl@extract</code>	102	<code>\tex_pdfobj:D</code>	504
<code>\ttl@setifthe</code>	3296	<code>\tex_prevdepth:D</code>	2555
<code>\ttl@setsubmark</code>	104, 3269, 3275	<code>\tex_topskip:D</code>	2549
<code>\ttl@settopmark</code>	104, 3263, 3294	<code>\tex_Uchar:D</code>	742, 753, 754
<code>\ttl@tocpart</code>	3239, 3241	<code>\tex_XeTeXcharglyph:D</code>	5133
<code>\ttlh@hang</code>	3203, 3213	<code>\tex_XeTeXglyphbounds:D</code>	5157
<code>\ttlh@runin</code>	3202, 3212	<code>\tex_xkanjiskip:D</code>	2061, 2065, 2075
<code>\tunec.def</code>	54	<code>\textasteriskcentered</code>	783
<code>\tw@</code>	450	<code>\the</code>	4822, 4825, 4826, 4845, 4849
<code>\uppercase</code>	49	<code>\thechapter</code>	2780, 2783, 2834, 3364, 3384, 3411, 3647
<code>\use@mathgroup</code>	1033, 1064	<code>\theparagraph</code>	3749
<code>\usepackage</code>	33, 33, 35	<code>\thepart</code>	2632, 2651, 2673, 2693, 3586
<code>\verbatim@font</code>	1712, 1716	<code>\thesection</code>	3348, 3355, 3371, 3378, 3387, 3395, 3402, 3418, 3666
<code>\vskip</code>	22	<code>\thesubparagraph</code>	3763
<code>\wrong@fontshape</code>	57	<code>\thesubsection</code>	3351, 3381, 3398, 3698
<code>\x@protect</code>	5474	<code>\thesubsubsection</code>	3732
<code>\xdef</code>	104	<code>\thispagestyle</code>	2612, 2613, 2750, 2761, 2763
<code>\XeTeXglyphbounds</code>	136	<code>\tiny</code>	4064, 4103
<code>\xkanjiskip</code>	81	tl commands:	
<code>\yahei</code>	8, 131	<code>\c_novalue_tl</code>	5343
<code>\youyuan</code>	8	<code>\c_space_tl</code>	508
<code>\z@</code>	710, 713, 2647, 2709, 2764, 2829, 2854, 2873, 2916, 2954, 2973, 2987, 3222, 3348, 3371, 3378, 3387, 3395, 3402, 3418, 4036, 4044, 4055, 4075, 4083, 4094	<code>\tl_clear:N</code>	219, 385, 1666, 1670, 1926, 1938, 2054, 2174, 2499, 2517, 2528, 3990, 5558
<code>\zhdig</code>	87	<code>\tl_clear_new:N</code>	60, 362
<code>\zhdigits</code>	28	<code>\tl_const:Nn</code>	3, 415, 420, 425, 430, 651, 673, 1081, 1094, 1422, 1829, 1831, 1832, 1833, 1840, 1841, 1842, 2193, 2421, 2427, 3826, 3845, 3847, 4720, 4723, 4724
<code>\zhnum</code>	87	<code>\tl_gput_right:Nn</code>	459, 485, 600, 1820, 1834, 1835, 1836, 1837, 1843, 1844, 1845, 5354, 5356, 5374
<code>\zhnumber</code>	28, 28	<code>\tl_gset:Nn</code>	248, 251, 254, 405, 557, 560, 1019, 1021, 1023, 1396, 1846, 1857, 4157, 4160, 4161, 4177, 4210
<code>\zihao</code>	27	<code>\tl_gset_eq:NN</code>	1878
<code>\ziju</code>	27	<code>\tl_gset_rescan:Nnn</code>	1403
<code>\zw</code>	80	<code>\tl_head:N</code>	3994
tex commands:		<code>\tl_head:n</code>	1447
<code>\tex_afterassignment:D</code>	733	<code>\tl_if_blank:nTF</code>	753, 1481, 1490, 1690, 1691, 1706, 2146, 2156
<code>\tex_chardef:D</code>	734	<code>\tl_if_empty:NTF</code>	401, 883, 2250, 2433, 3538, 4168
<code>\tex_def:D</code>	5550	<code>\tl_if_empty:nTF</code>	154, 1753, 1767, 1780, 5521
<code>\tex_dimexpr:D</code>	150, 2232	<code>\tl_if_eq:NNTF</code>	1853, 2186, 3905
<code>\tex_divide:D</code>	84, 2233, 2273	<code>\tl_if_exist:NTF</code> ..	117, 599, 1828, 1831, 1832, 1833, 1840, 1841, 1842, 2347, 2378, 2389, 2392, 3838, 3928, 5373
<code>\tex_edef:D</code>	5604	<code>\tl_if_in:NnTF</code>	5585
<code>\tex_endlinechar:D</code>	92, 97, 109, 545	<code>\tl_if_novalue:nTF</code>	1312, 1333, 1684, 2498
<code>\tex_font:D</code>	507, 5129	<code>\tl_map_inline:Nn</code>	1596
<code>\tex_glueexpr:D</code>	2223	<code>\tl_map_inline:nn</code>	3991
<code>\tex_hangindent:D</code>	3020	<code>\tl_new:N</code>	111, 155, 157, 474, 599, 736, 885, 886, 974, 1078, 1129, 1156, 1173, 1174, 1195, 1538, 1672, 1811, 1818, 1871, 2068, 2194, 2207, 2349, 2419, 2420, 3095, 3096, 5367, 5368, 5373, 5561, 5562, 5563
<code>\tex_iftrue:D</code>	3844	<code>\tl_put_left:Nn</code>	2027, 5390
<code>\tex_ignorespaces:D</code>	1185, 1344, 1349, 2116, 2257, 3935, 5360		
<code>\tex_immediate:D</code>	504		
<code>\tex_indent:D</code>	2540		
<code>\tex_interlinepenalty:D</code>	2529		
<code>\tex_kanjiskip:D</code>	1958, 1972, 1986		
<code>\tex_noindent:D</code>	2530, 3018		
<code>\tex_numexpr:D</code>	754		
<code>\tex_pagegoal:D</code>	2548		

`\tl_put_right:Nn` 369, 1025,
 1679, 2015, 2016, 2020, 2449, 2488, 2490, 3994, 3995, 5384
`\tl_remove_all:Nn` 1448
`\tl_replace_all:Nnn` 1494
`\tl_replace_once:Nnn` 1821
`\tl_rescan:nn` 145, 5603, 5612
`\tl_set:Nn`
 . 145, 214, 503, 535, 536, 537, 553, 732, 887, 930, 975,
 1075, 1076, 1115, 1178, 1190, 1332, 1491, 1493, 1534,
 1873, 2005, 2053, 2069, 2130, 2170, 2208, 2260, 2350,
 2497, 2500, 2504, 3199, 3218, 3237, 3242, 3250, 3904, 4122
`\tl_set_eq:NN`
 .. 892, 893, 944, 1020, 1022, 1179, 1204, 1634, 1646, 2195
`\tl_set_rescan:Nnn` 5559
`\tl_tail:N` 3204
`\tl_tail:n` 3214
`\tl_to_str:n` 153, 5502, 5531, 5572, 5573
`\tl_trim_spaces:n` 1493
`\tl_trim_spaces_apply:nN` 1663
`\tl_use:N` 461
`tocdepth` 25, 3490
`\today` 2325, 2330, 2333, 2338
`today` 11, 2325
token commands:
`\c_parameter_token` 105, 3327, 3329, 3332
`\token_to_meaning:N` 5516
`\token_to_str:N` 60, 179, 760, 1002,
 1010, 1019, 1021, 1023, 1232, 1233, 1234, 1236, 1243,
 1244, 1245, 1307, 2375, 2376, 3943, 5136, 5313, 5332, 5420
`\toks` 4822, 4823, 4825, 4826, 4845, 4849
`\topsep` 4047, 4058, 4086, 4097
`\TrimSpaces` 2144
`\ttdefault` 1863
`\ttfamily` 1826
`\twocolumn` 2755
`\typeout` 2780, 2781, 3237

U

`ubuntu` 35
um commands:
`\um_input_math_symbol_table:` 1057, 1061
`\um_sym:nnn` 1060
um internal commands:
`__um_input_math_symbol_table:` 1069
`__um_switchto_literal:` 1066
`__um_sym:nnn` 1068
`\undefined` 4847
`\UnicodeEncodingName` 760
`\unless` 2917, 2955, 2974
`\unskip` 2990
`\upshape` 1743, 1746
use commands:
`\use:N` 894, 936, 937, 983, 1019, 1021, 1023, 1099,
 1148, 1647, 2572, 2578, 2587, 3004, 3067, 5586, 5606, 5620
`\use:n` 158, 982,
 1083, 1262, 1315, 1321, 1393, 1397, 1496, 1872, 3013,
 3042, 3320, 3836, 3953, 4182, 5337, 5455, 5497, 5548, 5569

`\use:nn` 5614
`\use_i:nn` ... 142, 997, 2566, 5401, 5406, 5426, 5431, 5441
`\use_i:nnn` 269
`\use_ii:nn` 143, 997, 2564, 2568, 5401, 5406, 5426, 5431, 5442
`\use_ii:nnn` 159, 274
`\use_iii:nnn` 279
`\use_none:n` ... 215, 220, 225, 961, 1164, 1396, 1531, 2311
`\use_none:nn` 520
`\usebeamercolor` 3432, 3451, 3470, 3629, 3689, 3723
`\usebeamerfont` 3431, 3442,
 3450, 3461, 3469, 3480, 3628, 3631, 3688, 3691, 3722, 3725
`\usefont` 1827
`\uselanguage` 3883, 3888, 3909, 3914
`\usepackage` 5313
`UTF8` 6, 34, 245

V

`\verse` 4132
`\vfil` 2620, 2742
`\vskip` 2661, 2664,
 2693, 2718, 2721, 2745, 2834, 2844, 2847, 2863, 2866,
 2976, 2979, 3433, 3452, 3471, 3612, 3630, 3650, 3690, 3724
`\vspace` 2623, 2824, 2827, 2849, 2852

W

`windows` 35
`windowsnew` 35
`windowsold` 35
`winfonts` 32
`\write` 4862

X

`\x` 443, 449, 453, 4824, 4827,
 4828, 4829, 4830, 4831, 4832, 4833, 4834, 4835, 4836,
 4837, 4838, 4839, 4840, 4841, 4842, 4843, 4850, 4860, 4864
x commands:
`\x:` 448
`\xdef` 4863
`\xeCJKsetup` 657, 1934, 2111, 2114, 2117, 2132

Y

`\yahei` 4791
`\youyuan` 4791

Z

`\zhdig` 2317
`\zhdigits` 28, 2324
`zhmap` 6, 32, 264
`\zhnum` 2316
zhnum commands:
`\zhnum_counter:n` 2308
`\zhnumber` 28, 2322
`\zhnumsetup` 2334, 2339
`\zhtoday` 2333, 2338
`\zihao` 27, 3934, 3943
`zihao` 8, 194
`\ziju` 27, 2256
`\zw` 711, 1955, 1993, 2070